FINAL REPORT

to

THE FLORIDA DEPARTMENT OF TRANSPORTATION
TRAFFIC ENGINEERING AND OPERATIONS OFFICE

On Project

"Development and Testing of Optimized Autonomous and Connected Vehicle Trajectories at Signalized Intersections"

FDOT Contract BDV31-977-45

UF | University *of* Florida
Transportation Institute

November 2017

Submitted by
The University of Florida

DISCLAIMER

The opinions, findings, and conclusions expressed in this publication are those of the authors and not necessarily those of the State of Florida Department of Transportation.

METRIC CONVERSION CHART

U.S. UNITS TO METRIC (SI) UNITS

LENGTH

SYMBOL	WHEN YOU KNOW	MULTIPLY BY	TO FIND	SYMBOL
in	inches	25.4	millimeters	mm
ft.	feet	0.305	meters	m
yd.	yards	0.914	meters	m
mi	miles	1.61	kilometers	km

METRIC (SI) UNITS TO U.S. UNITS

LENGTH

SYMBOL	WHEN YOU KNOW	MULTIPLY BY	TO FIND	SYMBOL
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft.
m	meters	1.09	yards	yd.
km	kilometers	0.621	miles	mi

Technical Report Documentation Page

1. Report No.

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle

Development and Testing of Optimized Autonomous and Connected Vehicle Trajectories at Signalized Intersections

5. Report Date

November 2017

6. Performing Organization Code

7. Author(s)

Lily Elefteriadou, Mahmoud Pourmehrab, Patrick Emami, Aschkan Omidvar, Clark Letter, Patrick Neal, Rebecca Kiriazis, Sanjay Ranka, Carl Crane, Shannon Ridgeway

8. Performing Organization Report No.

9. Performing Organization Name and Address

University of Florida Transportation Institute
512 Weil Hall, PO Box 116580
Gainesville, FL 32611-6580

10. Work Unit No. (TRAIS)

11. Contract or Grant No.

FDOT Contract BDV31-977-45

12. Sponsoring Agency Name and Address

Florida Department of Transportation
605 Suwannee Street, MS 30
Tallahassee, FL 32399

13. Type of Report and Period Covered

Final Report (August 2015 – November 2017)

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Significant improvements in automated and connected vehicle technologies are expected to create a revolution in how we move and move things. Automated vehicles can operate using a variety of sensors such as GPS, lidar, radar, and smart cameras, as well as terrain information, and they have the ability to communicate with infrastructure as well as surrounding vehicles. The objectives of this research were to develop, test, and deploy an intelligent real-time intersection traffic control system in order to optimize simultaneously signal control and automated vehicle trajectories when the traffic stream consists of autonomous, connected, and conventional vehicles. The system developed was first simulated in MATLAB. Simulation showed that the proposed system can minimize the total travel time at an isolated intersection.

The system developed was also tested at the Florida Department of Transportation's Traffic Engineering Research Laboratory (FDOT TERL) facility. This report provides an overview of the hardware and software developed for the project, including a local server, DSRC (dedicated short range communications) receiver for the server, interface to the signal controller, sensor fusion system, radio communication software and hardware for vehicle to infrastructure communications. The system was tested under different scenarios. The outputs and video footage (http://avian.essie.ufl.edu/gallery/) showed that the system is capable of providing optimal trajectories to automated vehicles in order to reduce delays. Future work should expand the algorithm to consider congested conditions, lane changing within the communications range, and the presence of pedestrians and bicycles.

17. Key Word

Automated vehicles, connected vehicles, signal control optimization

18. Distribution Statement

No restrictions

19. Security Classif. (of this report)

Unclassified

20. Security Classif. (of this page)

Unclassified

21. No. of Pages

71

22. Price

EXECUTIVE SUMMARY

Significant improvements in automated and connected vehicle technologies are expected to create a revolution in how we move and move things. Automated vehicles can operate using a variety of sensors such as GPS, lidar, radar, and smart cameras, as well as terrain information, and they have the ability to communicate with infrastructure as well as surrounding vehicles. It is highly likely that in the not too distant future, connected and autonomous vehicle will be operating side by side in large numbers, along with conventional vehicles. The objectives of this research were to develop, test, and deploy an intelligent real-time intersection traffic control system in order to optimize simultaneously signal control and automated vehicle trajectories, considering the presence of autonomous, connected, and conventional vehicles in the traffic stream.

The system developed was first simulated in MATLAB. A total of 3,000 scenarios was tested to consider varying demand levels, communication ranges, automated vehicle percentage, and saturation headways. The results showed that for lower saturation headways, the average travel time decreases as the automated vehicle percentage increases because since these vehicles can more easily follow shorter headways. On average, lower effective greens are allocated to higher demand levels. In these cases, there are more requests from conflicting movements to switch the right of way as the demand increases, while the signal can be extended without any interruption for the low demand scenarios. The same pattern can be seen for varying communication ranges: the higher the communication range, the better the ability to design a platoon of trajectories ahead of time, which can assign longer green intervals with fewer interruptions.

The system was then implemented and tested at the Traffic Engineering Research Laboratory (TERL) facility. This report provides an overview of the hardware and software developed for the project, including a local server, DSRC (dedicated short range communications) receiver for the server, interface to the signal controller, sensor fusion system, radio communication software, and hardware for vehicle to infrastructure communications. Testing at the TERL was conducted under various scenarios. The outputs and video footage (http://avian.essie.ufl.edu/gallery/) showed that the system is capable of providing optimal trajectories to automated vehicles in order to reduce delays.

Future work should expand the algorithm to consider congested conditions, lane changing, and the presence of pedestrians and bicycles.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

1.1 Background

Significant improvements in autonomous and other advanced vehicle technologies as well as their connectivity and interaction with future generation traffic systems may significantly affect traffic safety and mobility. Autonomous vehicles are those that have the ability to operate on their own and do not require a human driver. Such vehicles are able to navigate the highway system through a variety of sensors, such as GPS, lidar, radar, and smart cameras, as well as terrain information. Connected vehicles are those that can communicate with infrastructure as well as surrounding vehicles through DSRC (dedicated short range communications), wifi, or cellular networks. The USDOT connected vehicle research program aims to enable wireless communications among vehicles, infrastructure, and personal communications devices (http://www.its.dot.gov/pilots/). The program focuses on pilot deployments in order to implement existing research concepts and encourage further innovation. Connected vehicle technology seeks to warn drivers of impending dangers while the vehicle is still controlled by a human driver. Automated vehicles are those that combine autonomy with connectivity.

It is highly likely that in the not too distant future, both connected and automated vehicles will be operating side-by-side in large numbers on our nation's highways, along with conventional vehicles. This creates many opportunities for improving surface transportation efficiency and safety. For example, the USDOT Multimodal Intelligent Traffic Signal Systems (MMITSS) initiative aims "to provide a comprehensive traffic information framework to service all modes of transportation, including general vehicles, transit, emergency vehicles, freight fleets, and pedestrians and bicyclists in a connected vehicle environment"[1].

According to the National Transportation Operations Coalition (NTOC, 2012), delays at traffic signals are estimated to be 5% to 10% of all traffic delay on major roadways and contribute an estimated 25% to the increase in total highway traffic delays during the past 20 years. Improvements in traffic signal timing have the potential to significantly benefit the transportation system. One source of delay at signals is inefficient green time utilization in response to fluctuating demand. Another source is driver reaction-related delays, including start-up delay. The use of autonomous and connected vehicle technology has the potential to reduce the impact of these two factors, through the use of its communication capability as well as the potential to fully control autonomous vehicle trajectories. Initiatives such as the MMITSS seek to optimize signal control in a connected vehicle environment and do not consider the potential of controlling automated vehicle trajectories. However, delay can be significantly reduced if the signal controller can develop optimal trajectories for automated vehicles and direct them to accelerate early (up to a maximum speed) or to modify their trajectory such that intersection capacity is fully utilized.

[1] https://www.its.dot.gov/research_archives/dma/bundle/mmitss_plan.htm first paragraph

1.2 Project Objectives

The main objective of this project is to enhance and expand a previously developed optimization algorithm (Li et al. 2014), and assemble, produce, and test the necessary software and hardware for enhancing traffic signal control operations simultaneously with vehicle trajectories, when the traffic stream consists of connected vehicles, autonomous vehicles, and conventional vehicles (i.e., those with no operating connectivity or automation.) The algorithm is capable of optimizing simultaneously vehicle trajectories of automated vehicles together with the signal control patterns at the intersection. The research team implemented the optimization algorithm at FDOT's Traffic Engineering Research Lab (TERL) which includes a signalized intersection in a closed course environment. A schematic overview of the proposed concept is shown in Figure 1-1.

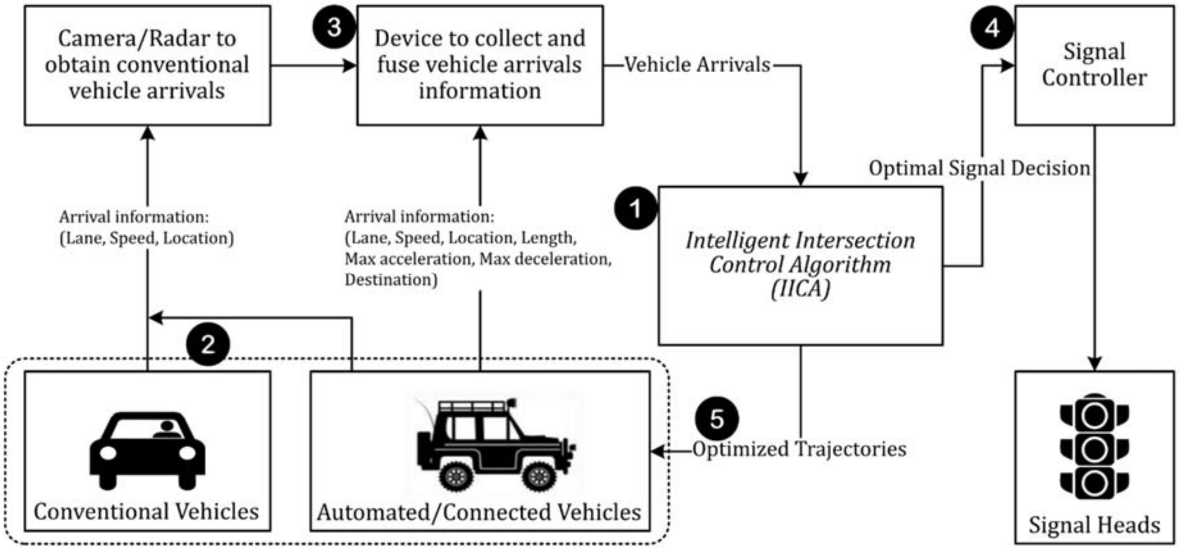

Camera/Radar to obtain conventional vehicle arrivals
3
Device to collect and fuse vehicle arrivals information
Vehicle Arrivals
4
Signal Controller
Optimal Signal Decision
Arrival information: (Lane, Speed, Location)
Arrival information: (Lane, Speed, Location, Length, Max acceleration, Max deceleration, Destination)
1
Intelligent Intersection Control Algorithm (IICA)
2
5
Optimized Trajectories
Conventional Vehicles
Automated/Connected Vehicles
Signal Heads

Figure 1-1 Overview of proposed concept

1.3 Organization

The next chapter provides an overview of the algorithm and its implementation at the TERL, while the third chapter discusses the equipment used. The fourth chapter describes the equipment used and the testing conducted at the TERL, along with the results and evaluation of the process. The fifth chapter summarizes the research conclusions and recommendations.

2 Optimization Algorithm Overview

The objective of the proposed algorithm is to maximize the intersection's throughput by jointly optimizing the automated vehicles' trajectories and the signalization pattern (phasing and timing). The algorithm is based on obtaining vehicle arrival information at a sufficient distance from the intersection (this is termed the available communications distance) so that optimal signal timings and autonomous vehicle trajectories can be obtained and transmitted back to the vehicles through the communications infrastructure.

A previous version of the optimization (developed by D. Zhuofei Li, a Ph.D. graduate of UF) was expanded to consider both automated and conventional vehicles. The algorithm was also revised to produce conventional signal timings that would be implementable in the field, rather than providing a reservation system that would be useable only by autonomous vehicles, which was the premise for the previously developed algorithm. Optimization is conducted to obtain both optimal signal timings and vehicle trajectories simultaneously. The algorithm developed was tested through simulation, and then implemented at the TERL.

A literature review was conducted early during this project to identify related research work and inform the development of this project. A summary of this literature review is provided in Appendix A.

2.1 Optimization Algorithm Input and Outputs

The algorithm inputs include the available communications range, the number of approaches and allowable phases, as well as minimum and maximum phase durations. Vehicle information obtained through Dedicated Short Range Communications (DSRC) includes the location and speed of the vehicle, as well as its entering lane. The algorithm produces optimal signal timings which are communicated to the signal controller, and optimized vehicle trajectories which are transmitted to each autonomous vehicle. The recommended trajectory for each vehicle is an ordered list containing pairs of location and time to be followed by the vehicle.

The following assumptions were used to develop and simulate the algorithm:

- The algorithm calculations and communication to the autonomous vehicles occur instantaneously, i.e., the calculation and communication time is negligible. This assumption was evaluated during the field test at the TERL.
- The conventional vehicle movement is simulated using the Gipps (1981) car-following model.
- No lane changing occurs once a vehicle arrival is detected.
- No pedestrians are present in the vicinity of the intersection, and no pedestrian phases are included in the algorithm.
- Demands are assumed to be below capacity and there are no oversaturated cycles.
- For simulation purposes, it is assumed that there is no data loss due to communication malfunctions.

- The intersection is located on level terrain with both left-turn and right-turn movements available to all vehicles.
- The algorithm uses a safe speed recommendation inside the communication area, however there might be cases the speed of individual vehicles is slightly higher at various points throughout the intersection communication area.

The algorithm described here was customized for the signalized intersection at the TERL, however it can be easily customized and used at other signalized intersections. It considers the presence of both automated and conventional vehicles, and it is currently simulated in MATLAB.

The algorithm is designed to be used as part of the intelligent intersection control system illustrated in Figure 1-1. The main purpose of the system is to use vehicle arrival information in advance to enhance the performance of an isolated intersection. In order to serve this purpose, vehicle arrival information is obtained at a certain communication distance. This arrival information is obtained from automated vehicles through DSRC communication, and from conventional vehicles through radar and/or video. The arrival information from different sources is fused so that the algorithm receives a unified set of arrival inputs. Then, the algorithm can generate the signal control decision and automated vehicles' trajectories. Finally, the required instructions are sent in a readable format to the automated vehicles and the procedure is continuously repeated.

2.2 Rolling Horizon Scheme

One of the important considerations in the algorithm development is the need to be able to obtain vehicle information early enough in order to be able to calculate and provide a meaningful trajectory to the stop bar. The allowable lag time (t_{Lag}) between when optimal signal control is obtained from the algorithm and when it will be implemented can be estimated based on the communications distance and the prevailing speeds. Based on this estimated lag time, a rolling horizon scheme is applied such that vehicle arrival data are obtained over an interval Δt_{Arr}, optimal trajectories and signal control are estimated at the end of that interval, and they are implemented through the subsequent interval Δt_{Opt}. Figure 2-1 illustrates the rolling horizon concept used in the algorithm implementation for one incoming lane (communication range is d_{Comm}.) The calculated signal timings are displayed after t_{Lag} time. Vehicle trajectories are communicated to the vehicles and implemented starting as soon as the calculations are complete.

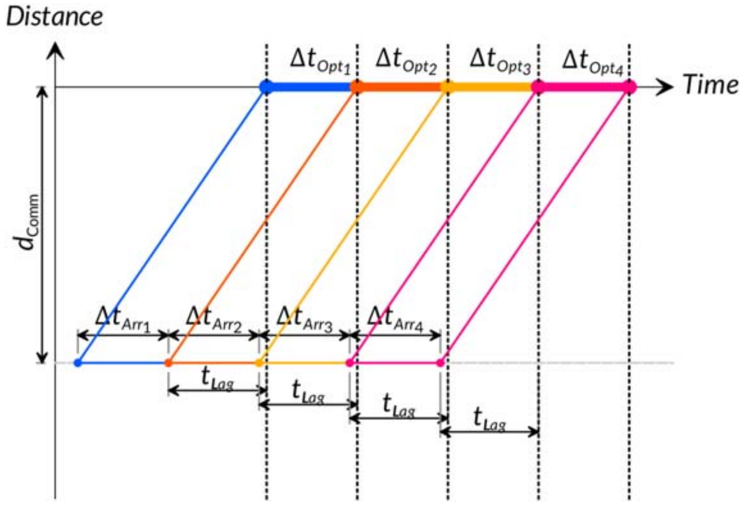


Figure 2-1 The rolling horizon scheme

A low value for t_{Lag} can result in insufficient time to provide full trajectories for all automated vehicles arriving during Δt_{Arr}. An appropriate estimation of the lag time can be obtained as a function of the communication range, the initial speed of vehicle, the maximum allowable speed, the vehicle acceleration/deceleration capability, and the crossing speed.

Example - t_{Lag} computation: Consider a four-leg intersection with six incoming lanes, and assume the communication ranges of 1000 feet and the maximum allowable speed of 40 mph on all lanes. Suppose vehicles' initial speed for all lanes follow the triangular distribution with a minimum of 34 mph and a vehicle is capable of maximum acceleration rate of 3 $\mathrm{ft/s^2}$ and maximum deceleration rate of -5 $\mathrm{ft/s^2}$ to adjust its speed.

According to the definition of t_{Lag}, it can be computed using a mathematical program (discussed later in this chapter):

$$t_{Lag} = \underset{l \in \{1,2,\dots,6\}}{\text{maximize}} \quad \underset{(V_2, a_1, a_3)}{\text{minimize}} \frac{1{,}000}{V_2} + \frac{\left(34 * \frac{5{,}280}{3{,}600} - V_2\right)^2}{2a_1V_2} - \frac{\left(V_2 - 40 * \frac{5{,}280}{3{,}600}\right)^2}{2a_3V_2}$$

$$subject\ to:$$

$$0 \leq V_2 \leq 40 * \frac{5{,}280}{3{,}600}$$

$$-5 \leq a_1 \leq 3$$

$$-5 \leq a_3 \leq 3$$

Using the solution method discussed in Appendix B, t_{Lag_l} for each lane is equal to 17.26 seconds. The slow vehicle detected at a 1000 feet distance from the stop bar needs 2.93 sec to accelerate from 34 mph to 40 mph. Next, the vehicle keeps the maximum speed all the way to the stop bar. In conclusion, the signal decision can be made about 17.26 sec earlier to guarantee that even the slowest first vehicle can reach the stop bar without any portion of the green interval be unutilized.

2.3 Optimization Frequency

Another issue addressed in the algorithm development relates to the frequency with which optimization is conducted. This is also equal to the duration of the interval over which vehicles are being detected, Δt_{Arr}, and optimal signal timings are estimated, Δt_{Opt}, as well as the timing of optimal vehicle trajectory transmission. As shown in Figure 2-1, the first vehicle within the interval Δt_{Arr} will have traveled some distance until the interval ends and optimal trajectories and signal timings are estimated and transmitted. If Δt_{Arr} is too long, the early vehicles will have already traveled a significant distance and would be too close to the intersection to make any meaningful trajectory adjustments. On the other hand, in order to optimize signal timings, we need Δt_{Arr} to be as large as possible so we can obtain several vehicle arrivals from all approaches and lanes before setting the signal timings.

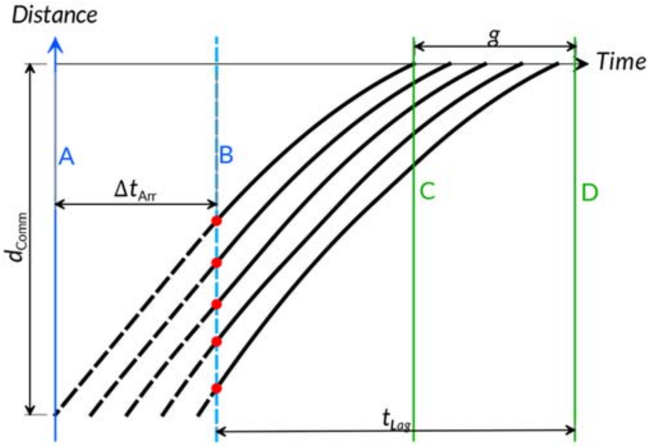


Figure 2-2 Optimal trajectories obtained at time B

Considering the vehicle keeps the entering speed as long as no trajectory is assigned, the traveled distance can be computed as:

$$\Delta d_{UC} = V_0 \, \Delta t_{Arr} \qquad \text{Equation 1}$$

where:

Δd_{UC} distance travelled without any trajectory assigned to the vehicle (Uncontrolled portion of trajectory).

V_0 initial speed at which vehicle speed and arrival time is collected

With the assumption that no queue is formed, the remaining distance to the stop bar should be adequate for the vehicle to safety decelerate to the recommended crossing speed. The following equation provides the lower bound on the communication range in order to achieve this:

$$d_{Comm} \geq V_0\, \Delta t_{Arr} + \frac{V_c^2 - V_0^2}{2a_{dec}} \qquad \text{Equation 2}$$

where:

V_c is the recommended maximum crossing speed at the stop bar.

The remaining variables are as defined before.

For instance, in order to safely decelerate a vehicle capable of -5 ft/s^2 deceleration rate to 15 mph if it enters the communication range at 40 mph, and its optimal trajectory is assigned after a second, at least 200 ft. of communication range is required. As the initial speed and Δt_{Arr} increase, the required communication range increases. This relationship is illustrated in Figure 2-3.

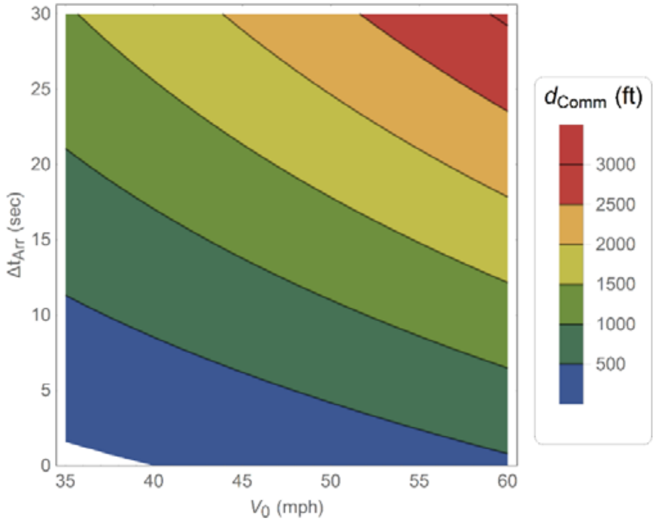


Figure 2-3 Feasible communication range region based on Δt_{Arr} and initial speed (a_{dec} = -5 ft/s^2,V_c = 40 mph)

2.4 Vehicle Trajectories Representation

The trajectory of each vehicle is represented by a set including pairs of time and the corresponding centerline distance to the stop bar as follows:

$$T_{ij} = \{(t_k, d_k)|\, vehicle\ j\ from\ lane\ i\ will\ be\ at\ distance\ equal\ to\ d_k\ from\ stopbar\ at\ time\ t_k\}$$

where:

i the lane index which belongs to the set of all inflow lanes I
j the vehicle index which represents the order of vehicles in lane i belongs to $J(i)$
k the index of trajectory points for vehicle j on lane i which belongs to $K(i,j)$

Using the denoted representation, the computed trajectory can be digitized as shown in Figure 2-4. In this fashion, detailed information describing the movement of vehicles is obtainable while numerical techniques for different derivatives can be used to extract speed, acceleration, or even jerk information. This approach can give enough flexibility to capture any sophisticated trajectory that no analytical function will fit well. This format is also compatible to the car-following models' output, which unifies the trajectory representation among all vehicle types.

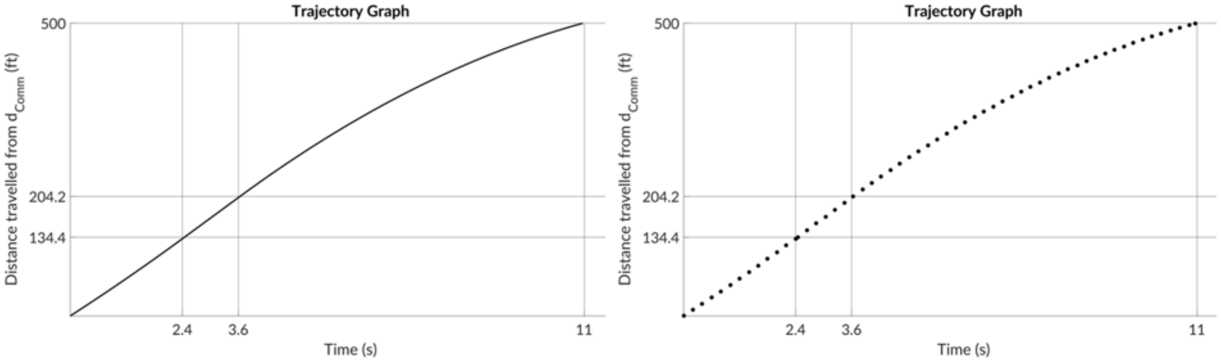


Figure 2-4 Trajectory representation as a set of scheduled time-distance pairs

2.5 Description of Traffic Engineering Research Lab (TERL) site

The TERL has a four-leg intersection with six inflow lanes and four discharge lanes. According to the intersection geometry shown in Figure 2-5, two approaches have exclusive left turn lanes. All the possible phases at the intersection are enumerated in Table 2-1. The objective of the proposed simulation is to determine the signal timing for these phases in a way that maximizes the throughput of the intersection.

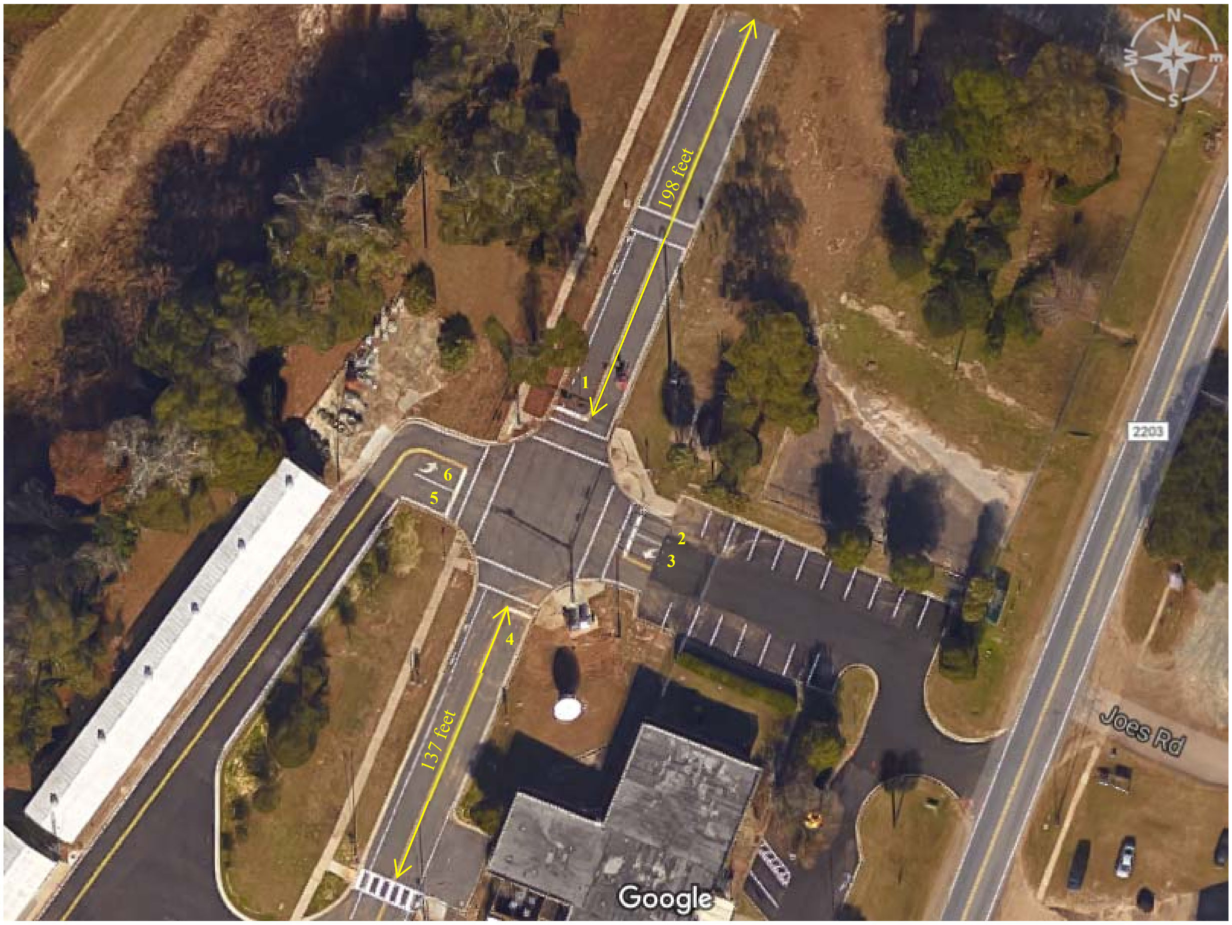


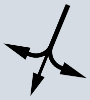
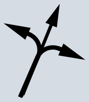
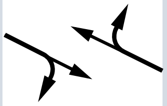
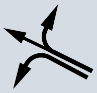
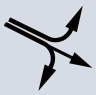
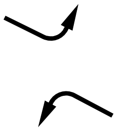
Figure 2-5 TERL intersection (inflow lanes are labeled as represented in the simulation)

2.6 Signalization

The algorithm uses pre-calculated yellow and all-red intervals and determines the green duration for each phase using predefined minimum green values. A minimum green interval of 5 seconds is assumed for all movements. These values can be modified as needed.

Table 2-1 includes all allowable phases, with each as a set of non-conflicting movements specifically for the TERL intersection. Of the six feasible phases, four phases are used in our algorithm (1, 2, 4, and 5). In the future, the algorithm can be extended to select a phase from all six feasible phases.

Table 2-1 Allowable phases and their corresponding non-conflicting movements for the TERL intersection

Phase No.	Φ1	Φ2	Φ3	Φ4	Φ5	Φ6
Lane(s)	1	4	2 and 5	2 and 3	5 and 6	3 and 6
Movements within the phase						

Assuming a mixed-traffic stream at the intersection, yellow and all-red intervals are required. For the purposes of simulation and testing, we assume 1.5 seconds of yellow and 1.5 seconds of all-red per phase. Figure 2-6 represents the association of each arrival interval and the corresponding implemented signal decision for a few consecutive phases. The left side of the figure shows the set of phases and the sequence they are implemented for this cycle. The decision on whether to extend a phase or switch to another phase is made as the vehicles' arrival is continuously monitored. The minimum green is used to prevent rapid switches. Each time interval there are new vehicle arrivals, these are communicated to the algorithm, and the optimal signalization pattern is determined. If the earliest arriving vehicle can be served within the projected phase (i.e., when the vehicle is projected to be arriving at the stop bar), the phase is allowed to be extended, up to the maximum green. Otherwise, the projected phase is terminated and a new phase, which serves the earliest arriving vehicles, is allocated minimum green. While this control is very similar to fully actuated signal control logic, the real-time information is obtained differently, and there is an opportunity to adjust the autonomous vehicle trajectories accordingly. The simulation results, which are provided later in this deliverable, show the functionality of the algorithm. As a result of the proposed process, no vehicle is forced to come to complete standstill and a smooth operation is anticipated.

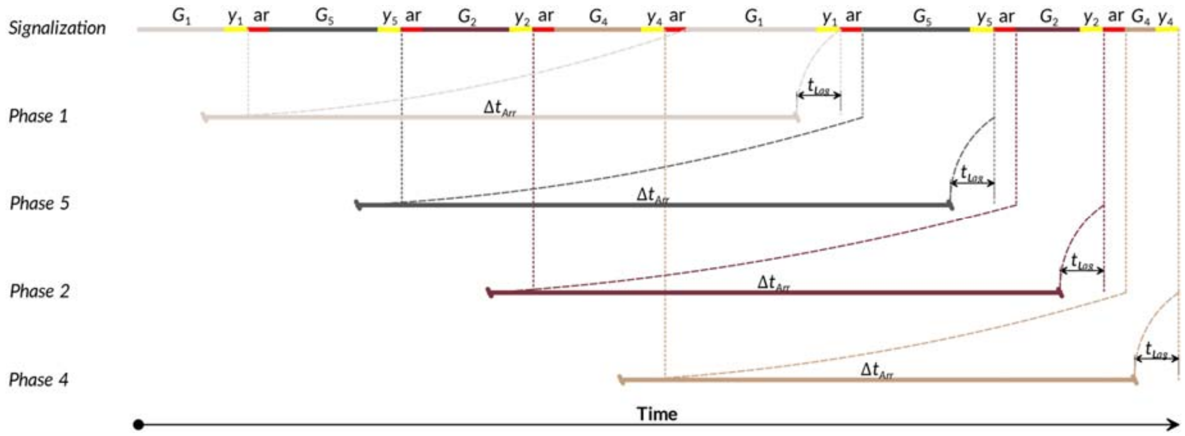

Signalization
G_1 y_1 ar G_5 y_5 ar G_2 y_2 ar G_4 y_4 ar G_1 y_1 ar G_5 y_5 ar G_2 y_2 ar G_4 y_4
Phase 1
Phase 5
Phase 2
Phase 4
Δt_{Arr}
t_{Los}
Time

Figure 2-6 Illustrative signal control plan for a four-phase, four-leg intersection

2.7 Trajectory Optimization and Estimation Process for Each Lane

The algorithm developed considers both autonomous and conventional vehicles. The arrival information for autonomous vehicles is obtained through the vehicles themselves, while the comparable information for conventional vehicles is obtained via radar (the equipment as well as the fusion of information from the two sources of data is described in the next chapter). Since conventional vehicles are not expected to follow a pre-specified trajectory, our simulation predicts their movement using a car-following model. We selected the Gipps (1981) car-following model, which has been implemented extensively, including in the commercially available microsimulator AIMSUN. The following subsections discuss each type of trajectory and its estimation in the simulation as well as in the algorithm used in the field.

2.8 Lead Vehicle Trajectory Optimization for Automated Vehicles

A lead automated vehicle is not constrained by other vehicles in front of it. Therefore, assuming no pedestrian presence or incident, the movement of the lead vehicle is guided by its desired speed and the signalization. Consequently, as proposed by Li et al. (2014), lead vehicles in all lanes can experience three trajectory components. The trajectory optimizer algorithm aims to design an adequate number of trajectory components to achieve the least travel time under the given signalization.

During the first component, the lead vehicle may need to accelerate/decelerate from its initial speed. During the next component, the vehicle may keep the appropriate speed for a given interval before the last component. During the third component, the vehicle may need to change speed to the maximum allowable crossing speed. Depending on its initial speed, the available distance to the stop bar, the maximum allowable crossing speed, and the time the green interval starts, any one of these three components may be eliminated to achieve the minimum possible travel time.

For the calculated trajectory to be feasible, components should be connected and no component should recommend any speed or acceleration/deceleration which exceeds the vehicle's capabilities or maximum allowable speed.

With the assumptions stated above, a mathematical program was formulated to minimize the total travel time of each vehicle. The travel time through each component can be computed using motion equations under either constant acceleration or constant speed as shown below:

$$t_1(V_2, a_1) = \frac{V_2 - V_0}{a_1} \qquad \text{Equation 3}$$

$$t_2(V_2, a_1, a_3) = \left(d_0 - \frac{V_2^2 - V_0^2}{2a_1} - \frac{V_3^2 - V_2^2}{2a_3}\right) / V_2 \qquad \text{Equation 4}$$

$$t_3(V_2, a_3) = \frac{V_3 - V_2}{a_3} \qquad \text{Equation 5}$$

where:

d_0 is the distance to the stop bar measured from the location the vehicle is first detected.

V_0 is the initial speed of the vehicle.

t_0 is the time the vehicle is first detected.

a_1 is the acceleration/deceleration rate each vehicle uses to change speed during the first trajectory component.

V_2 is the speed during the second trajectory component.

V_3 is the final speed at the stop bar (assumed to be equal to the maximum crossing speed V_c).

a_3 is the acceleration/deceleration rate vehicle use to change speed in third component.

Summing all travel times for all components of the lead vehicle, after some mathematical manipulations, the total travel time from detection point to the stop bar can be represented by the following equation:

$$T(V_2, a_1, a_3) = \frac{d_0}{V_2} + \frac{(V_0 - V_2)^2}{2a_1V_2} - \frac{(V_2 - V_3)^2}{2a_3V_2} \qquad \text{Equation 6}$$

On that foundation, the Lead Vehicle Trajectory Optimization (LTO) program can be formulated to minimize the lead vehicle's travel time. The LTO objective is to minimize the lead vehicle's total travel time subject to the available green time, speed, and acceleration/deceleration constraints. The set of variables needed to be determined by solving LTO includes the acceleration/deceleration rate through the first and third components and the speed during the second trajectory component. The LTO program is as follows:

$$\underset{(V_2, a_1, a_3)}{\text{minimize}}\ T(V_2, a_1, a_3) \qquad LTO$$

$$0 \leq T(V_2, a_1, a_3) - t_{G_i} \leq t_{p_i}$$

$$0 \leq V_2 \leq V^{max}$$

$$a^{min} \leq a_1 \leq a^{max}$$

$$a^{min} \leq a_3 \leq a^{max}$$

where:

p_i represents phase $i \in \{1, \dots, n\}$ where n is the total number of phases; this phase is the one scheduled to serve the subject vehicle

t_{G_i} is the time when the green interval for phase i begins.

t_{p_i} is the duration of green for phase i.

V^{max} is the maximum allowable speed within the communication range.

a^{min} is the maximum deceleration rate of the subject vehicle.

a^{max} is the maximum acceleration rate of the subject vehicle.

All other parameters and variables are as defined before.

The only drawback in using the mathematical program LTO is that the optimization belongs to a non-convex class which can be sophisticated to solve even with very few variables and equations. However, another representation of the problem can approximate it into a single variable constrained optimization problem which can be solved very efficiently (Appendix B).

2.9 Lead Vehicle Trajectory Estimation for Conventional Vehicles

Unlike the lead automated vehicle, a lead conventional vehicle behaves according to the driver, environment, and the vehicle's capabilities. Therefore, the trajectory optimization approach cannot provide an optimal movement. In our simulations we assume that the lead conventional vehicle would desire to maintain its speed as recorded when entering the communication range.

2.10 Follower Vehicle Trajectory Optimization for Automated Vehicles

Automated vehicle trajectories are optimized so that they arrive during the green interval at the maximum possible speed (crossing speed) and with the minimum headway at the stop bar. Assuming no congestion and queue at the intersection, the full trajectory can be communicated to the vehicle as indicated earlier.

The concept developed by Li et al. (2014) was enhanced and used here. First, the optimal hypothetical trajectory to assure the vehicle crosses the intersection with a minimum headway and maximum allowable discharge speed is constructed. Next, the algorithm searches for a safe possible adjustment of the vehicle's entering speed in order to match the hypothetical trajectory estimated. If any feasible speed change is found, the vehicle's trajectory will be the combination of an adjustment component, augmented by the remaining portion of the best hypothetical trajectory. This way the vehicle can arrive at the stop bar at the minimum headway and maximum discharge headway possible. If a feasible speed adjustment cannot be determined, the algorithm estimates the vehicle's movement independent from the lead vehicle. Then the earliest arrival at maximum discharge speed will be scheduled as suggested for the lead vehicle trajectory.

2.11 Follower Vehicle Trajectory Estimation for Conventional Vehicles

During the field implementation, a radar-based device is used to obtain the lane, location and speed of conventional vehicles once they get inside the communication boundary. Estimating and anticipating the follower vehicle's trajectory is important because it not only affects the signal decision, but the vehicle can be the lead vehicle to another conventional or automated vehicle behind them. If a conventional vehicle is leading another conventional vehicle, the follower will be assumed to operate according to the car-following model. If the conventional vehicle is leading an automated vehicle, its movement can directly affect the estimation of the automated vehicle's optimal trajectory.

Therefore, the behavior of conventional vehicles must be estimated in the simulation as they affect the decisions for automated vehicles' trajectories and signalization. The conventional follower vehicle is restricted by both the vehicles in front of it and the upcoming signal.

2.12 Simulation Process

To simulate our algorithm and evaluate it prior to field implementation, we replicated the TERL intersection in the computer, and assumed that traffic arrives upstream of each of the lanes at the communication range entry point. The inter-arrival distribution used in simulation for each lane follows the exponential distribution with its parameter equal to the lane's average headway. The initial speed of vehicles is assumed to follow the triangular distribution. The vehicle type is a binary parameter assigning each vehicle as conventional or automated, and it uses a user-specified percentage for each type. Other generic constants used are reported in Table 2-2.

The algorithm is coded in MATLAB to implement the flowchart shown in Figure 2-7. The algorithm starts with generating traffic for all lanes. As shown, signal control is determined jointly with the trajectory optimization. The trajectory optimization process is detailed in the bottom part of the graph.

Table 2-2 Simulation parameters

Parameters	Value	Note
Traffic generation duration	15 minutes	The simulation terminates when all the generated vehicles are served.
Communication range	Varies in each scenario starting from 500 feet to 3000 feet.	For real-world testing the range is determined based on available communication technology.
Connected/Automated vehicles percentage	Varies over different scenarios. Ranging from 30% to 100%.	The rest of the traffic stream is composed of conventional vehicles.
Average headway at the communication range in each lane	Varies over different scenarios. Ranging from 8 sec to 60 sec.	Assuming arrival times follow Poisson distribution, the inter-arrivals are randomly generated from negative exponential.
Saturation headway	2, 1.5, 1 sec	Corresponding to 450, 600, 900 veh/hr/ln thresholds for low-demand levels.

Arriving speed at the communication range in each lane	Follows triangular distribution with following parameters:
	Minimum = 0.85 * 40 = 34 mph
	Peak = 40 mph
	Maximum = 1.10 * 40 = 44 mph	

Maximum allowable speed within the communication range	40 mph	
Crossing speed at the stop bar	Depending on the movement: 40 mph for through movements 30 mph for right or left turns	
Length of vehicle	15 feet	
Maximum acceleration rate	10 ft/s^2 (about g/3)	
Maximum deceleration rate	-15 ft/s^2	
Desired speed for Gipps car-following model	40 mph	It is set same as the maximum allowable speed of automated vehicles.
Minimum green time for each phase	4.6 sec	Values are set to be consistent to the saturation headway of 1800 veh/hr.
Yellow time for each phase	1.5 sec	
All-red time for each phase	1.5 sec	
Trajectory points' time difference	1 sec	This gives the resolution of computed trajectories.

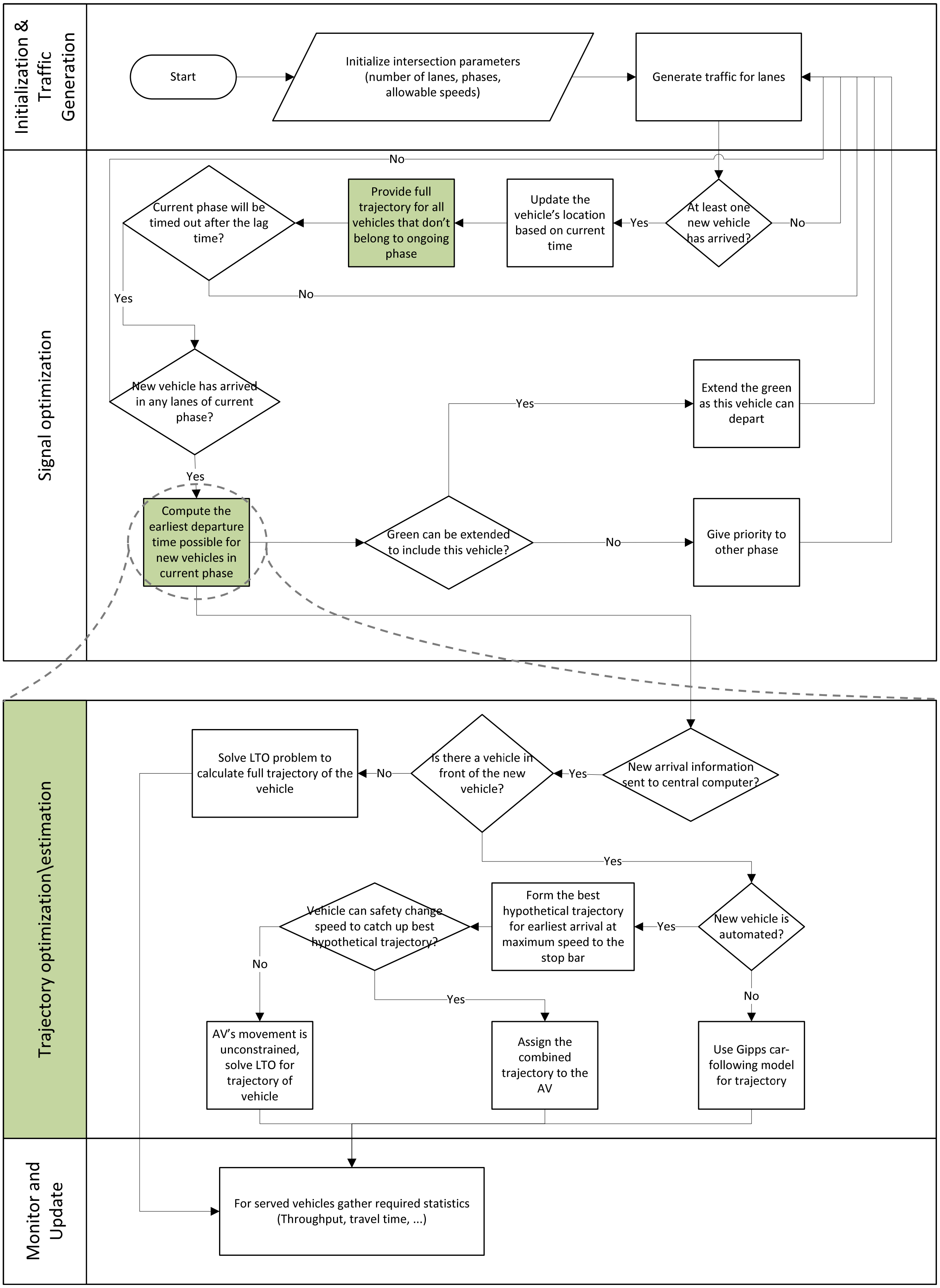

Initialization & Traffic Generation
Start
Initialize intersection parameters (number of lanes, phases, allowable speeds)
Generate traffic for lanes
Signal optimization
No
At least one new vehicle has arrived?
Yes
No
Update the vehicle's location based on current time
Provide full trajectory for all vehicles that don't belong to ongoing phase
Current phase will be timed out after the lag time?
No
Yes
New vehicle has arrived in any lanes of current phase?
Yes
Compute the earliest departure time possible for new vehicles in current phase
Green can be extended to include this vehicle?
Yes
Extend the green as this vehicle can depart
No
Give priority to other phase
Trajectory optimization\estimation
New arrival information sent to central computer?
Yes
Is there a vehicle in front of the new vehicle?
No
Solve LTO problem to calculate full trajectory of the vehicle
Yes
New vehicle is automated?
Yes
Form the best hypothetical trajectory for earliest arrival at maximum speed to the stop bar
Vehicle can safety change speed to catch up best hypothetical trajectory?
No
AV's movement is unconstrained, solve LTO for trajectory of vehicle
Yes
Assign the combined trajectory to the AV
No
Use Gipps car-following model for trajectory
Monitor and Update
For served vehicles gather required statistics (Throughput, travel time, ...)

Figure 2-7 Simulation process for intersection control algorithm

2.13 Simulation Results

This section presents the results of the simulation process using the TERL intersection layout. A variety of scenarios exploring the effect of communication range, connected/automated vehicles' percentage in traffic stream, and demand level was examined. The values tested were selected considering the possible scenarios to be tested at the TERL.

Figure 2-8 displays trajectories of incoming vehicles and the respective signalization after 60 seconds of simulation, by lane. As shown, lanes 2 and 3 or lanes 5 and 6 have the same signalization, as they are assigned to the same phase. The blue vertical line indicates the time in the simulation when the trajectories of new incoming vehicles are computed, therefore, as expected, each figure demonstrates the future optimal plan in every lane.

Figure 2-9 shows the arrival and departure curves for each lane. The time difference between the arrival and departure curves represents the total travel time of the vehicle from the communication range to the stop bar. Vehicles may arrive at any time; however, they may depart only when their phase is assigned the right of way. The vertical distance between arrivals and departures indicates the number of vehicles within the communication range.

Figure 2-10 provides the same information (i.e., arrivals and departures) for the entire 15 minutes of simulation to illustrate how the optimization is able to handle the demand for the entire simulation period.

Figure 2-11 illustrates the travel time distribution by lane, measured from the time a vehicle enters the communication range until it crosses the stop bar. Similarly, Figure 2-12 indicates the delay distribution for each lane at the end of a 15-minute simulation. Delay for a single vehicle is computed as the free flow travel time of the vehicle subtracted from its actual travel time once it crosses the stop bar.

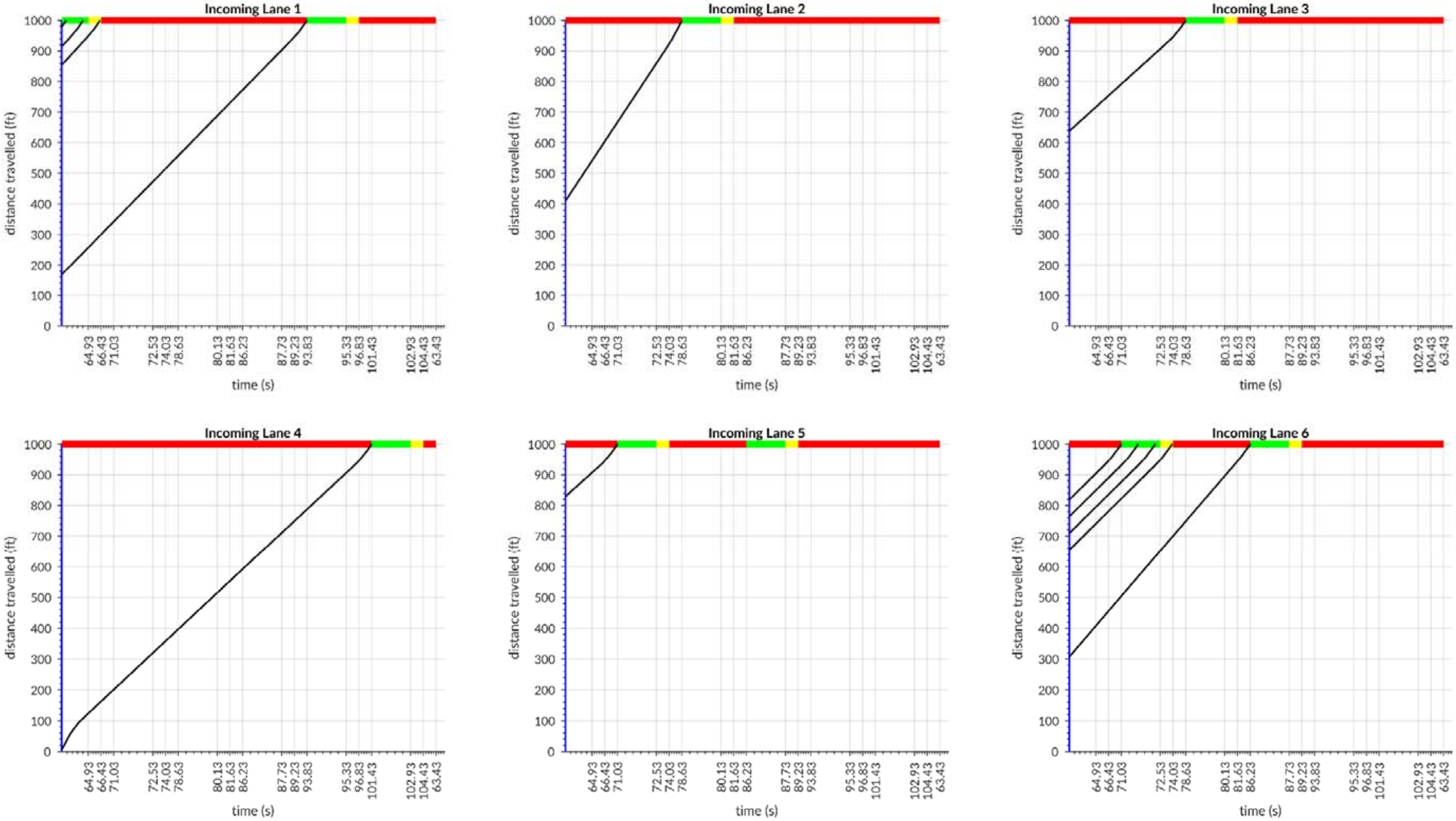


Figure 2-8 Trajectories and signalization after 60 seconds of simulation in each lane

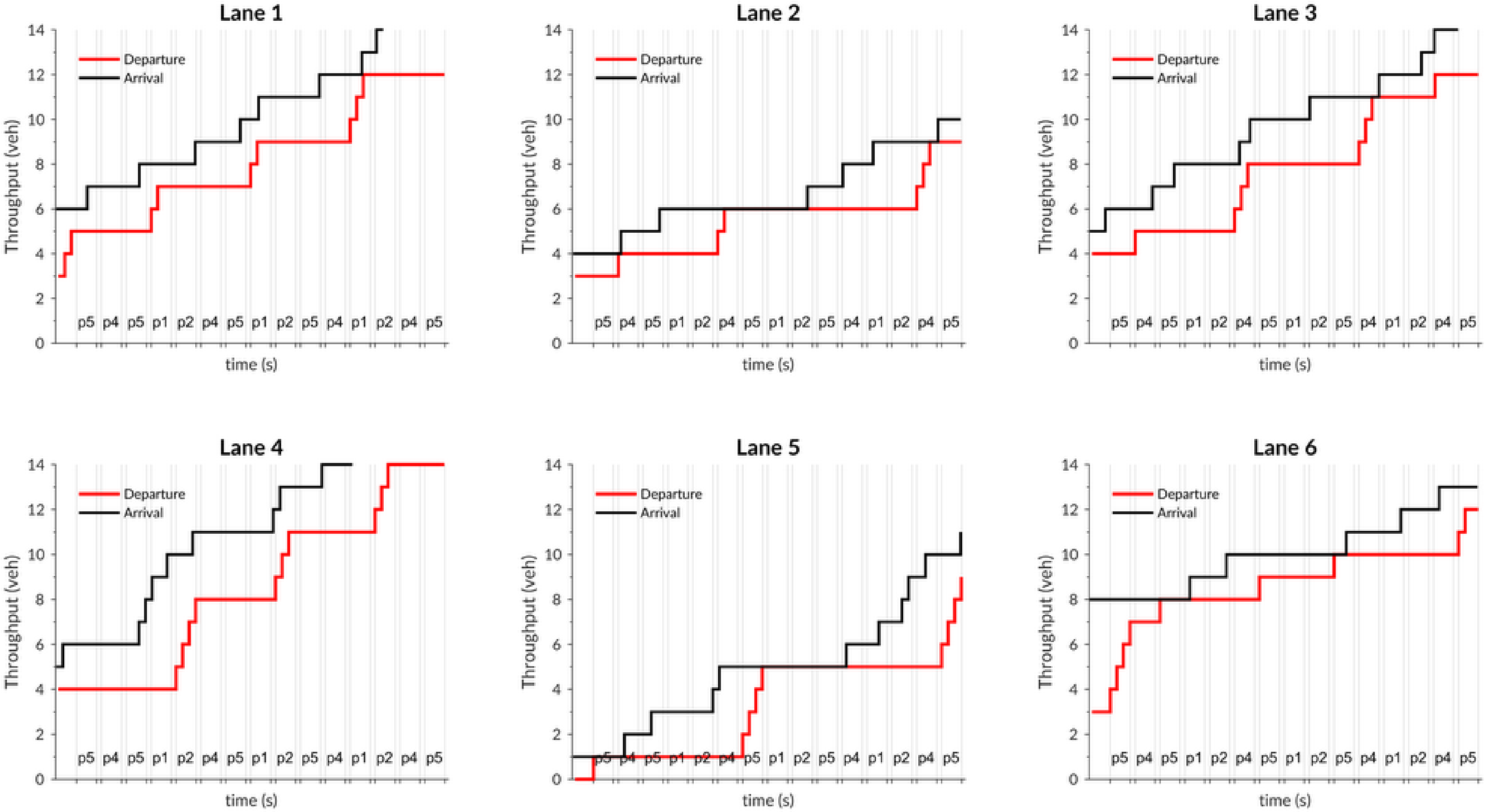


Figure 2-9 Departure/throughput curves from time = 60 sec to time = 180 sec during simulation, by phase, p(i)

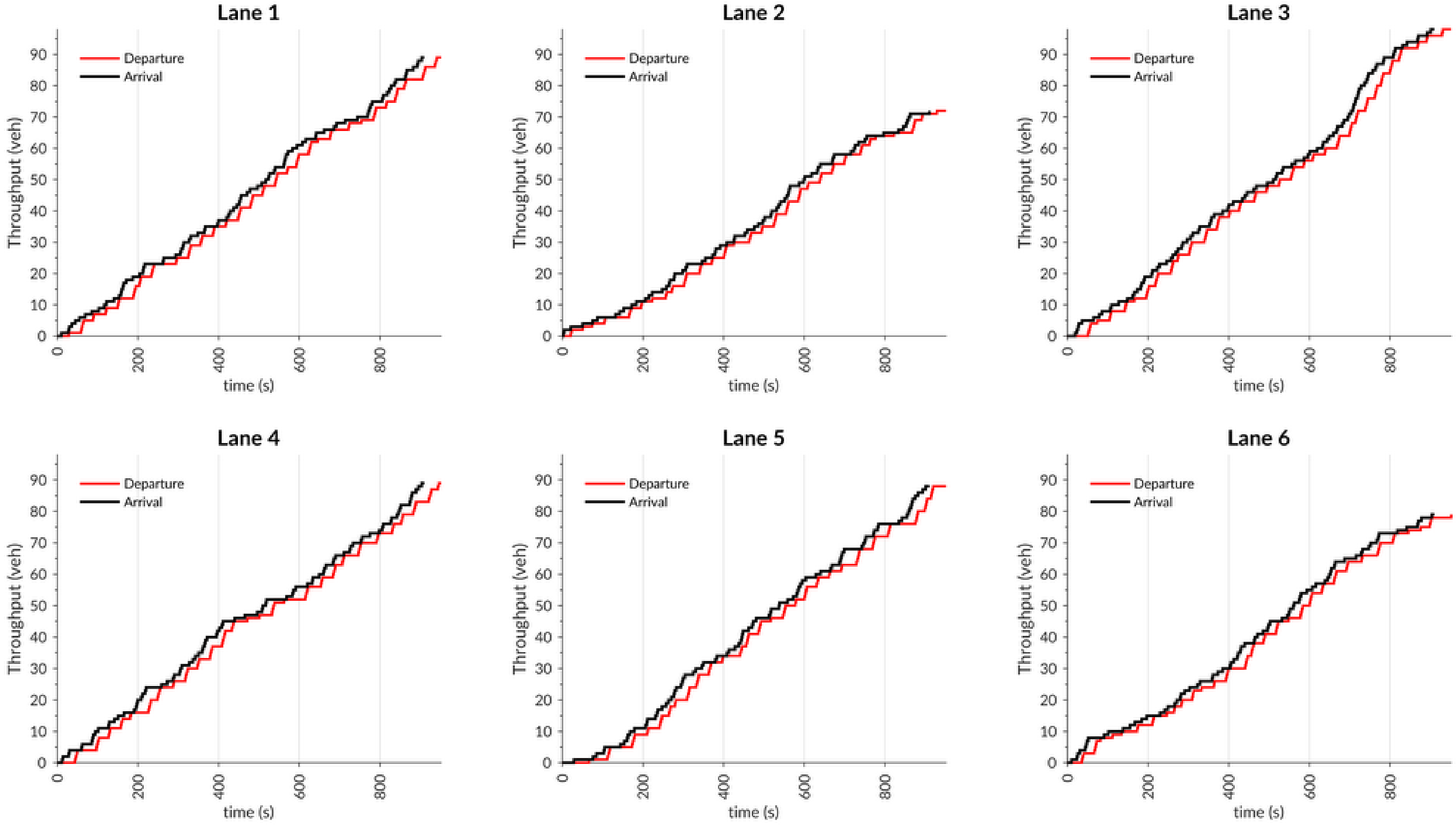


Figure 2-10 Departure/throughput curves for 15 minutes of simulation

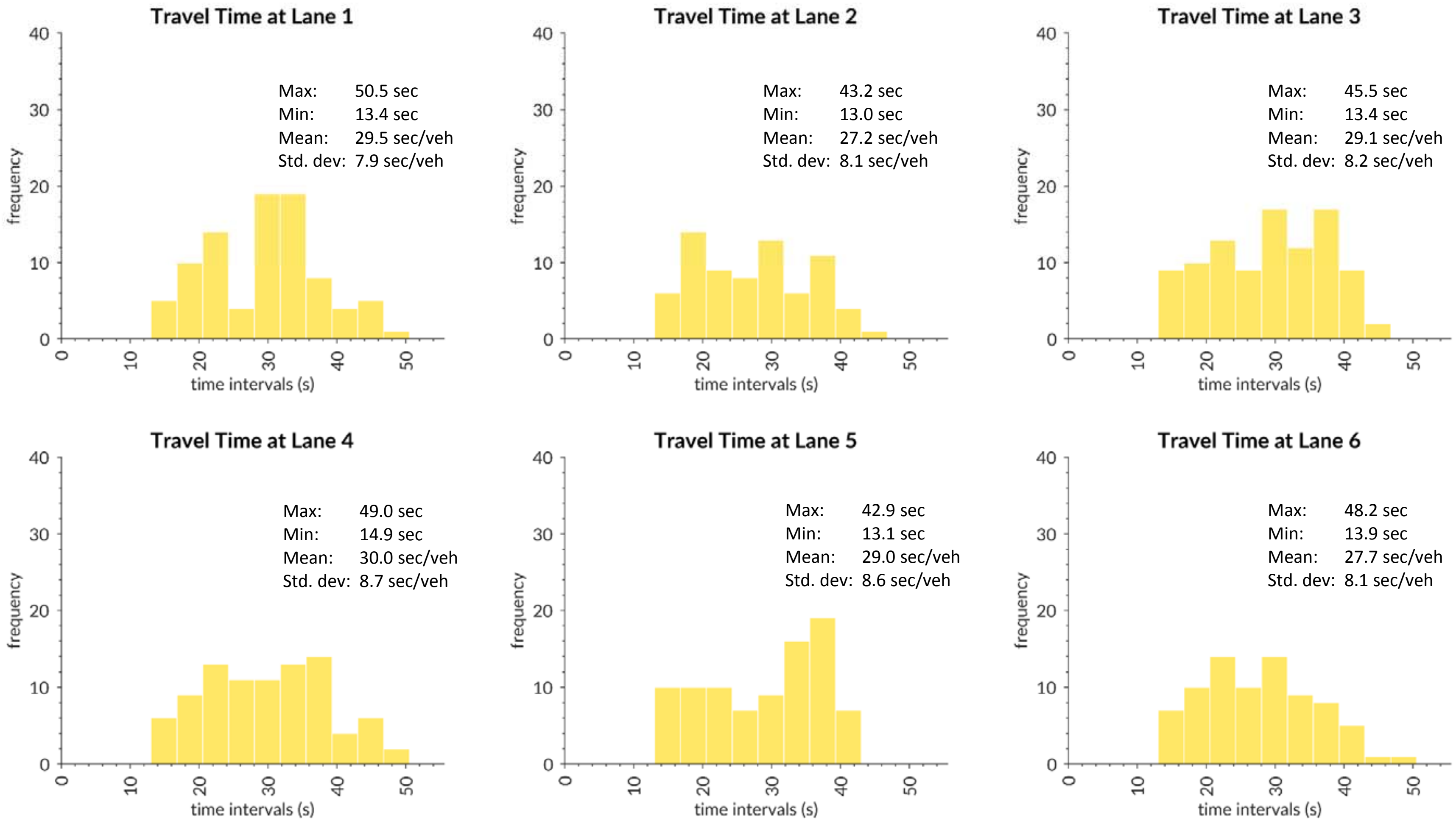


Figure 2-11 Travel time distribution for each lane for 15 minutes of simulation

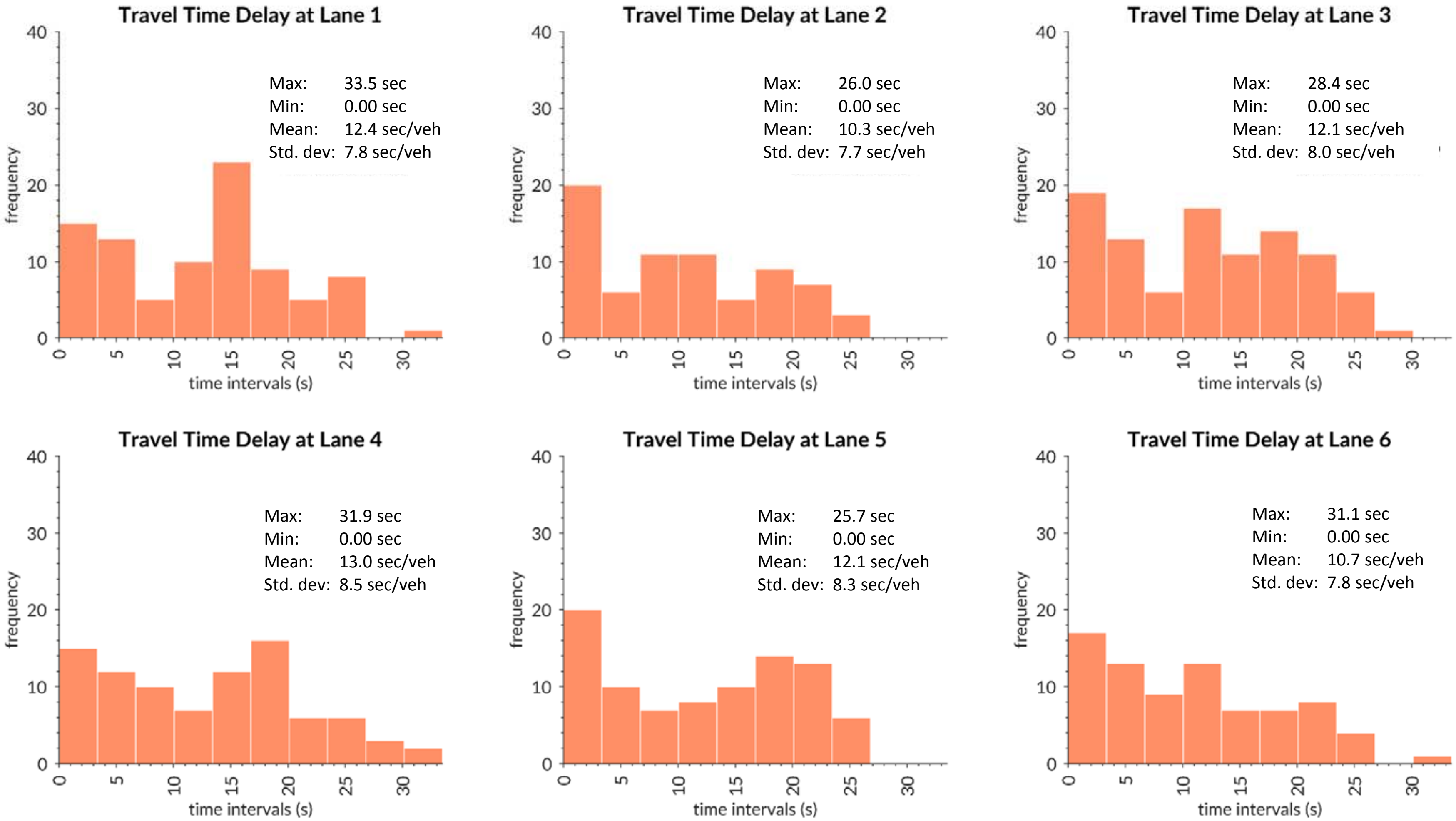


Figure 2-12 Delay distribution for each lane for 15 minutes of simulation

Several scenarios were tested to investigate the amount of improvement based on different control variables. The following list of control variables are generated to examine a variety of situations:

Average flow rate
The average time headway is used, assuming a negative exponential distribution, to generate the traffic.

The average flow rate, in veh/hr/ln, can be computed as the inverse of the average time headway and indicates the demand level by lane.

d_{Comm}
The average communication range, in feet, from the stop bar. This indicates the location where vehicles are first detected and trajectories are computed.

AV %
The percentage of connected/automated vehicles in the traffic stream. The remaining portion of traffic consists of conventional vehicles.

Saturation headway
The minimum time headway at which vehicles can cross the stop bar. This parameter affects only the automated vehicles' movement as conventional vehicles follow the Gipps car following model's trajectory.

To study the effect on the performance of the intersection, three outcome variables were used:

Average travel time
The average travel time, in seconds, considering all vehicles from all approaches.

The travel time for a vehicle is measured from the time it is detected to the time it crosses the stop bar in its lane.

Average travel time delay
The average travel time delay, in seconds/vehicle, measured by subtracting the free flow travel time from the actual recorded travel time of each vehicle.

Average effective green
The average effective green interval, in seconds, is the average of actual green intervals plus the yellow interval allocated to a particular phase within the simulation period.

A total of 3000 scenarios were tested (ten cases of the demand level, the communication range, and the automated vehicle percentage, and three cases of saturation headways.) The evaluation of all scenarios is illustrated in Figure 2-13, 14, and 15.

Figure 2-13 demonstrates the effect of all four control variables on the average travel time. The most noticeable trend is the sharp increase in the average travel time for the higher demand rates. As the travel time for each vehicle is measured from the communication range to the stop bar, travel time increases as the communication range increases. For lower saturation headways, the average travel time decreases as the AV percentage increases, since AVs can more easily follow shorter headways.

Because of the strong correlation between average travel time and the average travel time delay, the same sensitivity to the control variables is evident in Figure 2-14. However, the average delay does not vary as much as travel time within each demand scenario, because the effect of the communication range on the travel time is removed. The same abrupt increase in delay, as for travel time, is observed for higher demand rates.

Figure 2-15 indicates the average length of effective green assigned to all phases after each simulation experiment. On average, lower effective greens are allocated to higher demand levels. In these cases, there are more requests from conflicting movements to switch the right of way as the demand increases, while the signal can be extended without any interruption for the low demand scenarios. The same pattern can be seen for varying communication ranges: the higher the communication range, the better the ability to design a platoon of trajectories ahead of time, which can assign longer green intervals with fewer interruptions.

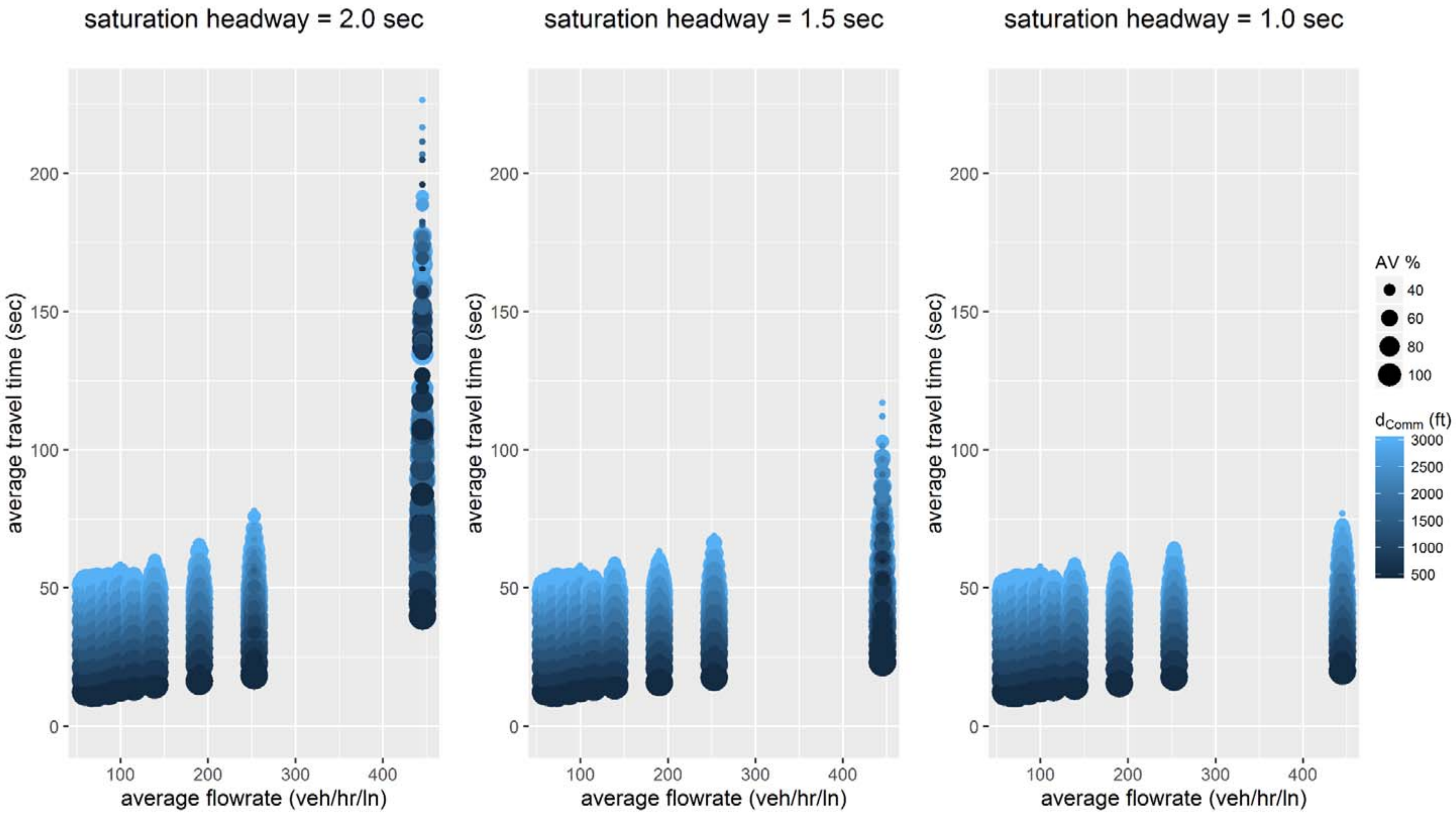


Figure 2-13 Average travel time as the simulation result for varying demand rate, communication range, and connected/automated vehicle's percentage

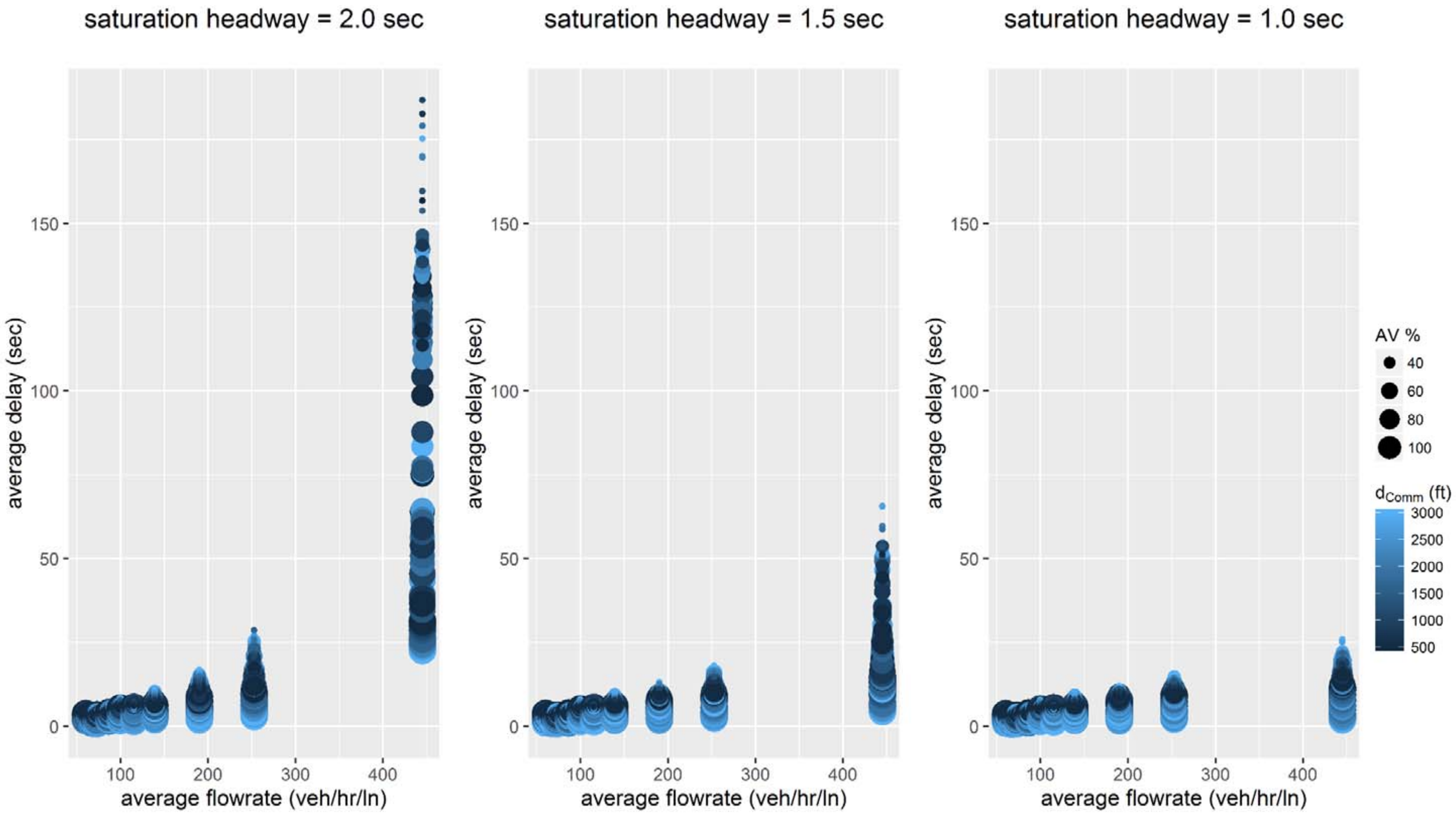


Figure 2-14 Average travel time delay as the simulation result for varying demand rate, communication range, and connected/automated vehicle's percentage

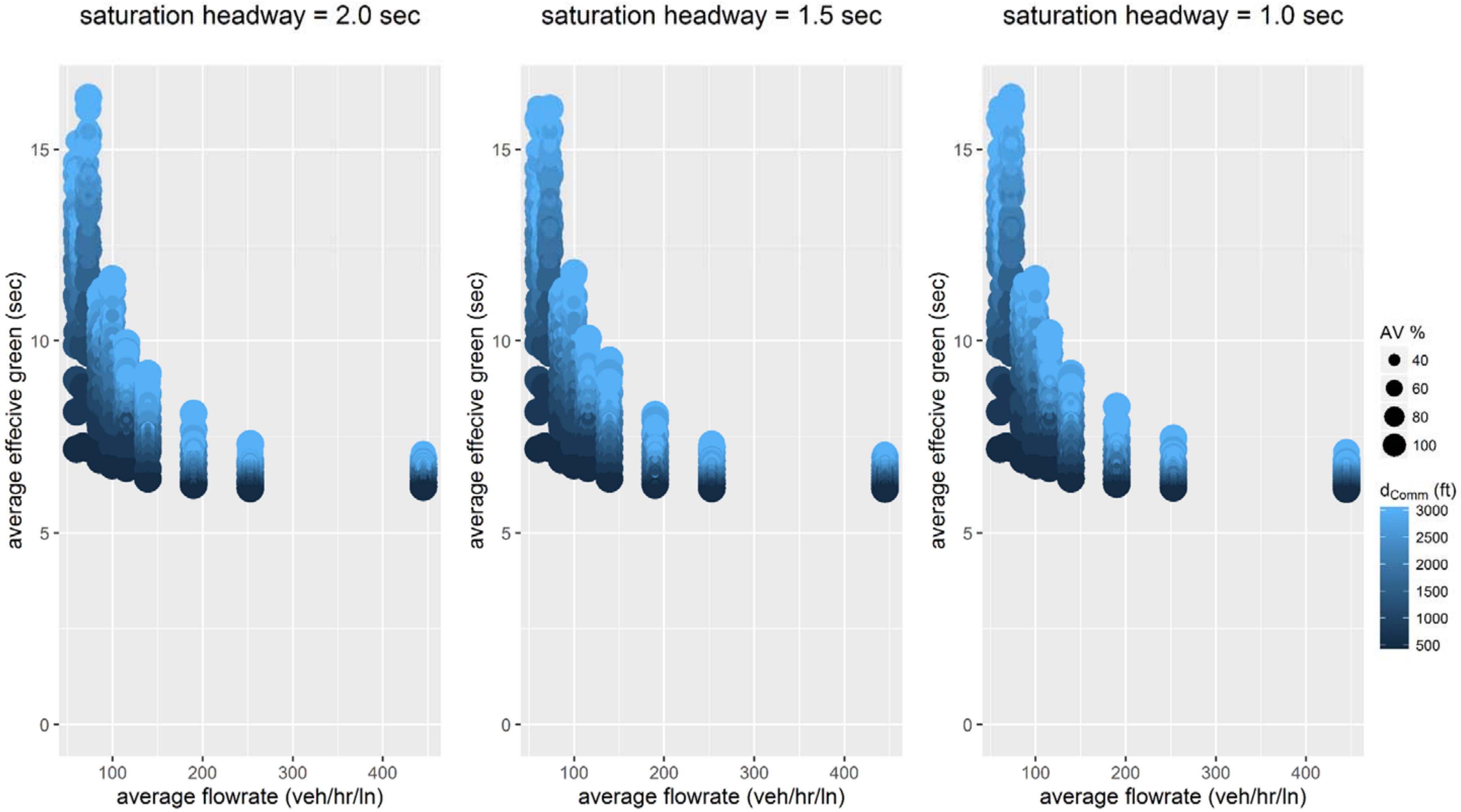


Figure 2-15 Average effective green as the simulation result for varying demand rate, communication range, and connected/automated vehicle's percentage

3 Development of Equipment Needs and Procurement

This chapter describes the equipment obtained and assembled for conducting field tests at the TERL. The following two (V2I) communication structures are used in the research:

- Vehicle to optimization algorithm: Each vehicle provides lane, speed, location (coordinates), destination (left, though, or right), vehicle length, acceleration/deceleration capabilities.
- Optimization algorithm to vehicle: Automated vehicles' trajectories are broadcasted back to each vehicle near instantaneously. The specific signal timing plan for the lane an automated vehicle is located is also be transmitted.

The following components were procured and assembled for testing of the complete system at the TERL:

1. Computer hardware needed to run the signal timing and trajectory optimization algorithms
2. Hardware needed to interface to the signal controller
3. Radar sensors used to identify conventional vehicles at the intersection
4. Radio hardware needed for communication between the connected and autonomous vehicles and the intersection computer
5. Hardware required for the connected vehicles to obtain their location and to interface with the driver
6. Hardware needed for the UF autonomous vehicle so that it can communicate with the system.

These components are depicted in Figure 3-1.

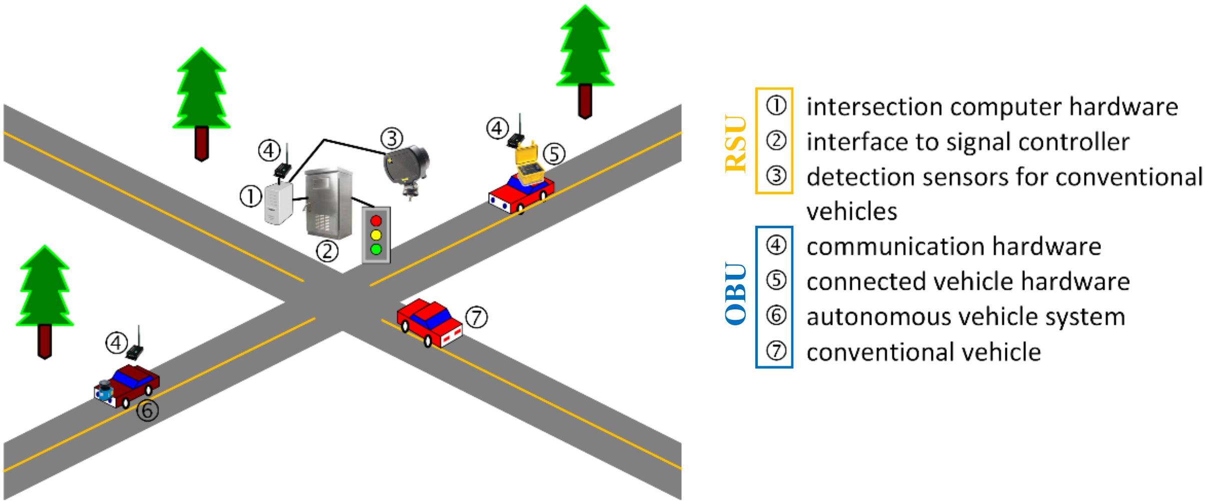

④
③
④
⑤
①
②
⑦
④
⑥
RSU
① intersection computer hardware
② interface to signal controller
③ detection sensors for conventional vehicles
OBU
④ communication hardware
⑤ connected vehicle hardware
⑥ autonomous vehicle system
⑦ conventional vehicle

Figure 3-1 System components

Components 1 to 3 are roadside units (RSU), while items 4 to 7 are related to vehicles and the equipment installed on them, referred as on-board unit (OBU). The remainder of this chapter discusses the detailed specifications and development of RSU and OBU components, as well as the connectivity among them.

3.1 Road Side Unit Components (RSU) – Infrastructure Elements

3.1.1 Intersection Computer Hardware

A desktop computer was assembled (Figure 3-2) to perform the optimization calculations and interface with the intersection controller and the DSRC Cohda radio. One of the Cohda radios was inserted into the external drive bay. Dual storage drives were used to allow fast read/write with the use of the solid state drive (SSD) and large storage space for log files on the hard disk drive (HDD)

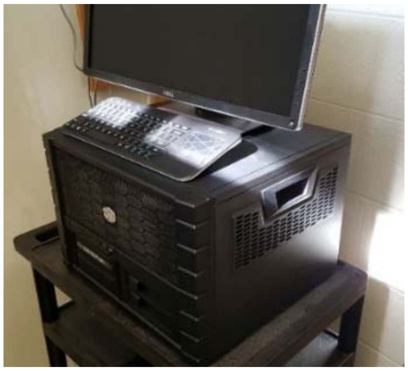


Figure 3-2 Desktop computer

• Motherboard	Supermicro Server Motherboard Dual LGA 2011 **Model#**: MBD-X10DRL-I-O
• CPU	Intel Xeon E5-2620 v3 Haswell 2.4 GHz **Model#**: BX80644E52620V3
• CPU Heatsink	Dynatron, Heatsink and Fan **Model#**: R17
• RAM	16GB (2x 8 GB Crucial PC4-17000 DDR4-2133MHz ECC) **Model#**: CT8G4RFS4213.18FA2
• Power Supply	750 W Thermaltake Toughpower **Model#**: PS-TPG-0750DPCGUS-1
• Hard drives	Intel 730 Series 2.5" 480GB SATA 6Gb/s Internal SSD **Model#**: SSDSC2BP480G4R5
• HDD	1TB Western Digital Black HDD **Model#**: WD1003FZEX
• Case	Cooler Master HAF XB EVO **Model#**: RC-902XB-KKN2
• Monitor	Dell Black 23" **Model#**: P2314H

The algorithm and the signal controller interface, discussed in the next section, were installed and run on this desktop computer.

3.1.2 Interface Hardware to the Signal Controller

The team has developed an interface to remotely control the actuated signal controller (ASC) from the computer. This interface receives the optimal signalization schema from the optimization algorithm and translates it into signal phasing and timing (SPaT) which is legible by the ASC. Through SNMP communication under NTCIP 1202 protocol, we establish the communication between the desktop and the ASC and with the maximum flexibility and accuracy, we access the Object Identifiers (OIDs) in the ASC and control the SPaT.

This actuated signal controller used in this research is National Electrical Manufacturers Association (NEMA) TS2 type 2 Econolite Cobalt ASC, donated to UF by Econolite. Cobalt is designed for the mobile computing environment. It fully meets the industry's ATC standard 5.2b and proposed standard 6.10. It is also designed to provide a combination of ATC controller open architecture functionality with the latest handheld technology and applications. The Linux-based operating system of this ASC makes programming and access to functions quite easy. The procedure starts with installing the controller inside a NEMA TS2 compatible cabinet with connector panel, conflict management units (CMU), switches and other standard components in a signal cabinet (Figure 3-3). Preliminary tests for connectivity were initially conducted at the UFTI's Signal Control Laboratory.

Generally, the communication between the SPaT device and the traffic signal controller (TSC) conforms to applicable NTCIP standards. However, in order to meet the requirements of the SPaT system, additional Simple Network Management Protocol (SNMP) objects have been added to the NTCIP 1202 Management Information Base (MIB). The Open System Interconnection (OSI) protocol stack describing the interconnection between the SPaT system and the TSC is listed as follows:

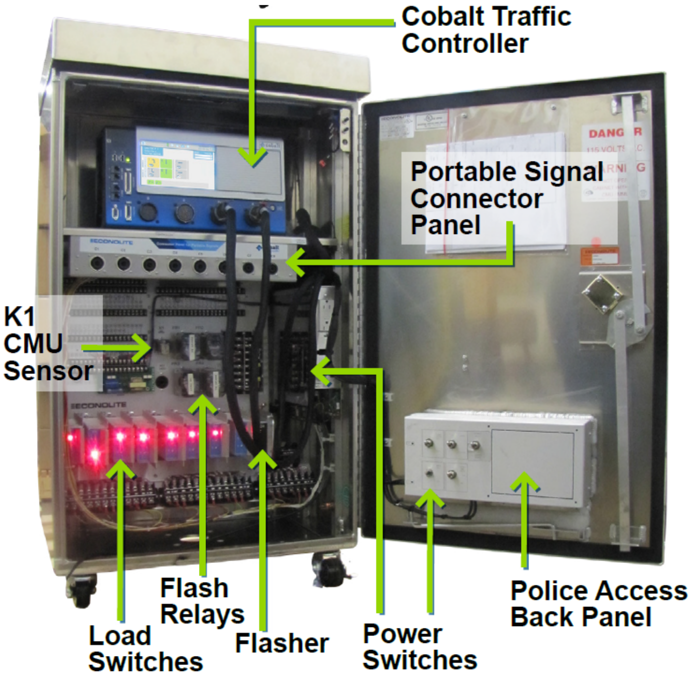


Figure 3-3 Signal Controller Inside Cabinet

• **Application**	**NTCIP data objects**
• Presentation	ASN.1
• Session	Sockets

• Transport	UDP
• Network	IPv4, port 6053
• Data Link	802.3 Ethernet
• Physical	10BASE-T

The interface between the SPaT system and the TSC makes use of SNMP data objects as well as a UDP/IP unicast message for the real-time SPaT status data. Each data element has on Object Identifier (OID) that corresponds to its physical location within the global naming tree. NTCIP OIDs are based on the root OID for the NEMA data objects. The OID for each data element in the SPaT system interface is prefixed by the NEMA object identifier as shown in Figure 3-4.

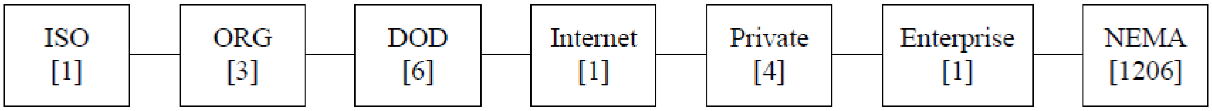


Figure 3-4 Root NEMA object identifier

The interface between the SPaT Interface Device or SOI and TSC makes use of the standard NTCIP 1202 objects along with the extended data objects. A snapshot of such OIDs are presented in Table 3-1. To manage the inherent complexities of the Cobalt controllers during the early Connected Vehicle research projects and pilot deployments, Econolite suggested to limit the scope of operations to the following commands:

General NTCIP Operation

- Fixed Time Vehicle Only
- Fixed Time with Ped
- Actuated Free
- Actuated Free with Ped

Coordinated

- Floating Force-Off/Fixed
- Transition Dwell/Smooth/Add
- Pattern Recalls Min/Max/Ped

Overlaps

- Included
- Lag Green, Yellow, Red
- Modifier (aka Not Included)

Table 3-1 NTCIP 1202 Interface SNMP Data Objects

Data Object	**OID**	**NTCIP 1202 Parag.**	**Get Individual OID**	**Get Group of OIDs**
phaseStatusGroupPedClears	NEMA.4.2.1.1.4.1.6.x	2.2.4.6	Yes	Yes
phaseStatusGroupWalks	NEMA.4.2.1.1.4.1.7.x	2.2.4.7	Yes	Yes
preemptControlTable	NEMA.4.2.1.6.3	2.7.3		
preemptControlEntry	NEMA.4.2.1.6.3.1	2.7.3		
preemptControlState	NEMA.4.2.1.6.3.1.2.x	2.7.3.2	Yes	
maxOverlaps	NEMA.4.2.1.9.1.0	2.10.1	Yes	

This set of

These commands will suffice to provide us the flexibility that we seek in order to control the operations of ASC from the server. Therefore, after we set the controller to send the push packet to a particular IP address (UDP Unicast) and when the communication has established to enable the SPaT unicast message in the controllers, we set the specific objects to enable sending and receiving data between the controller and the server. We make sure that every time we command an updated SPaT, the overlap systems are not violated and the controller avoids unnecessary all-red or flashing phases. All software is developed with Python in the Linux Ubuntu 14 platform.

3.1.3 Detection Sensors for Conventional Vehicles

As discussed earlier, conventional vehicles are detected through radar technology. The intelligent intersection control system's optimization component relies on the availability of vehicle arrival information to best serve the incoming traffic. Thus, we use a radar sensor to detect and classify all approaching vehicles. Only autonomous and connected vehicles are able to communicate directly with the intersection through DSRC radio communication. While this provides the arrival information for these two classes of vehicles, the controller will need to rely on the radar sensor to emulate the necessary information from conventional vehicles that are not equipped with a radio transmitter.

The radar sensor generates a basic safety message (BSM) for each vehicle, including location and speed of the vehicle. The BSM data obtained from the radar sensor are then fused with the DSRC radio BSMs that the intersection receives from autonomous and connected vehicles. We developed a discriminative algorithm informed by the fused data to classify incoming traffic as being from the conventional, connected, or autonomous vehicle classes. This algorithm is further discussed in the next chapter.

The radar sensor is able to initially detect vehicles at a distance of up to 600 feet. Upon detection, it provides the controller with the vehicle's speed and distance from the stop bar within the lane it is traveling in. Due to natural factors such as occlusion, weather, and range limitations, the speed and location data obtained from this sensor may be slightly inaccurate. This is a Doppler-based radar that relies on motion to detect objects; hence, detecting and tracking stopped or slow-moving vehicles will also provide a challenge for the radar sensor. However, as conventional vehicles approach the intersection, the accuracy at which the radar sensor can track these vehicles will improve; characterizing this change in uncertainty will be investigated in future work. For this project, our goal is to use the radar for the detection of conventional vehicles to provide the optimization algorithm with sufficient information to carry out trajectory optimization for the autonomous vehicles. We evaluate the performance of the radar sensor with the initial detection information and its impact on the uncertainty of location and speed of detected vehicles.

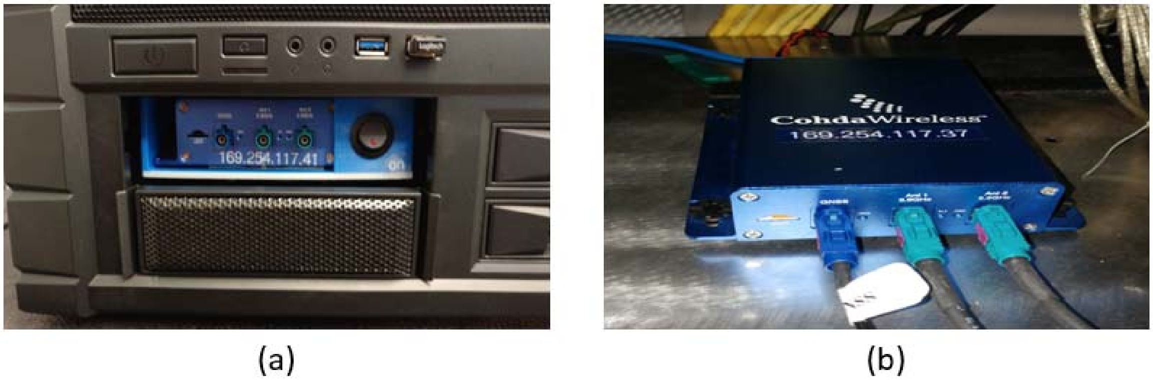
The development and testing of the radar sensor is being handled by Image Sensing Systems (ISS), a company that specializes in developing intelligent traffic sensors. ISS has field tested the radar to characterize its performance on many operational conditions with different congestion levels.

In summary, we procured a desktop computer which is equipped with communication hardware and software to enable sending and receiving data to/from the signal controller and radar. This computer enables us to communicate efficiently with the OBU equipment. The computer collects conventional vehicle data through radar, SPaT from signal controller, and messages through DSRC from connected and autonomous vehicles, processes them and sends feedback to OBU and the signal controller.

3.2 Road Side Unit Components (RSU)—Vehicle Elements

3.2.1 Communication Hardware

Every connected vehicle has a Cohda MK5 radio, as seen in Figure 3-5, in order to be able to transmit and receive DSRC messages. The intersection has a radio mounted inside the computer running the control optimization algorithm. It is able to receive the connected safety message (CSM) sent by the vehicles and forward it to the optimizer. The intersection radio then receives an intersection approach message (IAM) from the optimizer and forwards it to a designated connected vehicle. This section describes the message structure of the CSM and IAM.

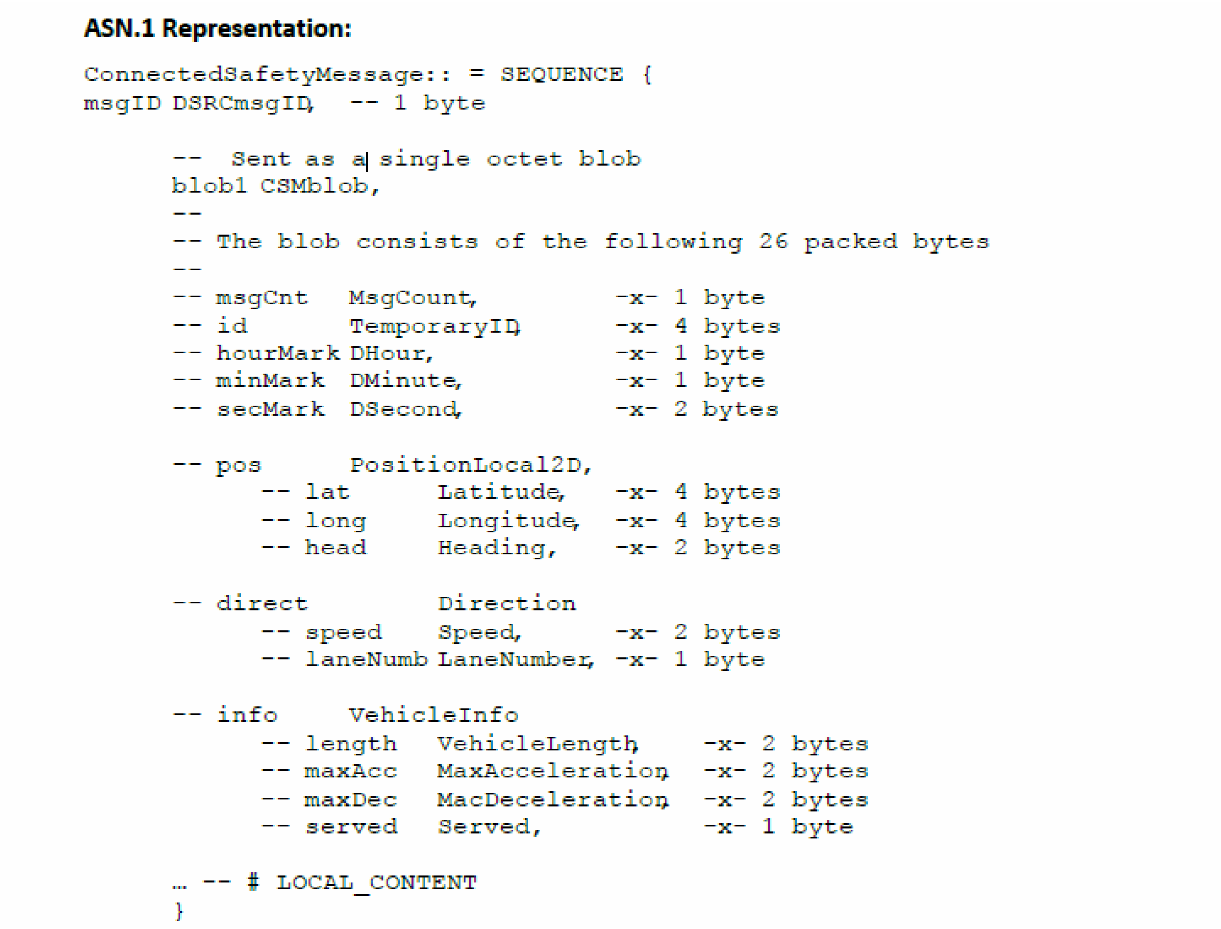


(a) (b)

Figure 3-5 Cohda MK5 located in the (a) intersection computer hardware, (b) autonomous vehicle

3.2.2 Connected Safety Message (CSM)

The CSM is sent out every 10 Hz by connected vehicle radio. Its first data element is its message ID number followed by the vehicle ID, the time that the message was sent, GPS coordinates, and direction/heading. Additional information in the message includes the vehicle's current speed, length, and its max acceleration and deceleration capabilities. It also sends a single byte that tells the optimizer what lane it is approaching in along with what direction it intends to follow as the vehicle passes through the intersection. The last byte tells the optimizer that it received the IAM sent by the intersection radio, at which point the connected vehicle will cease transmitting the CSM (Figure 3-6).



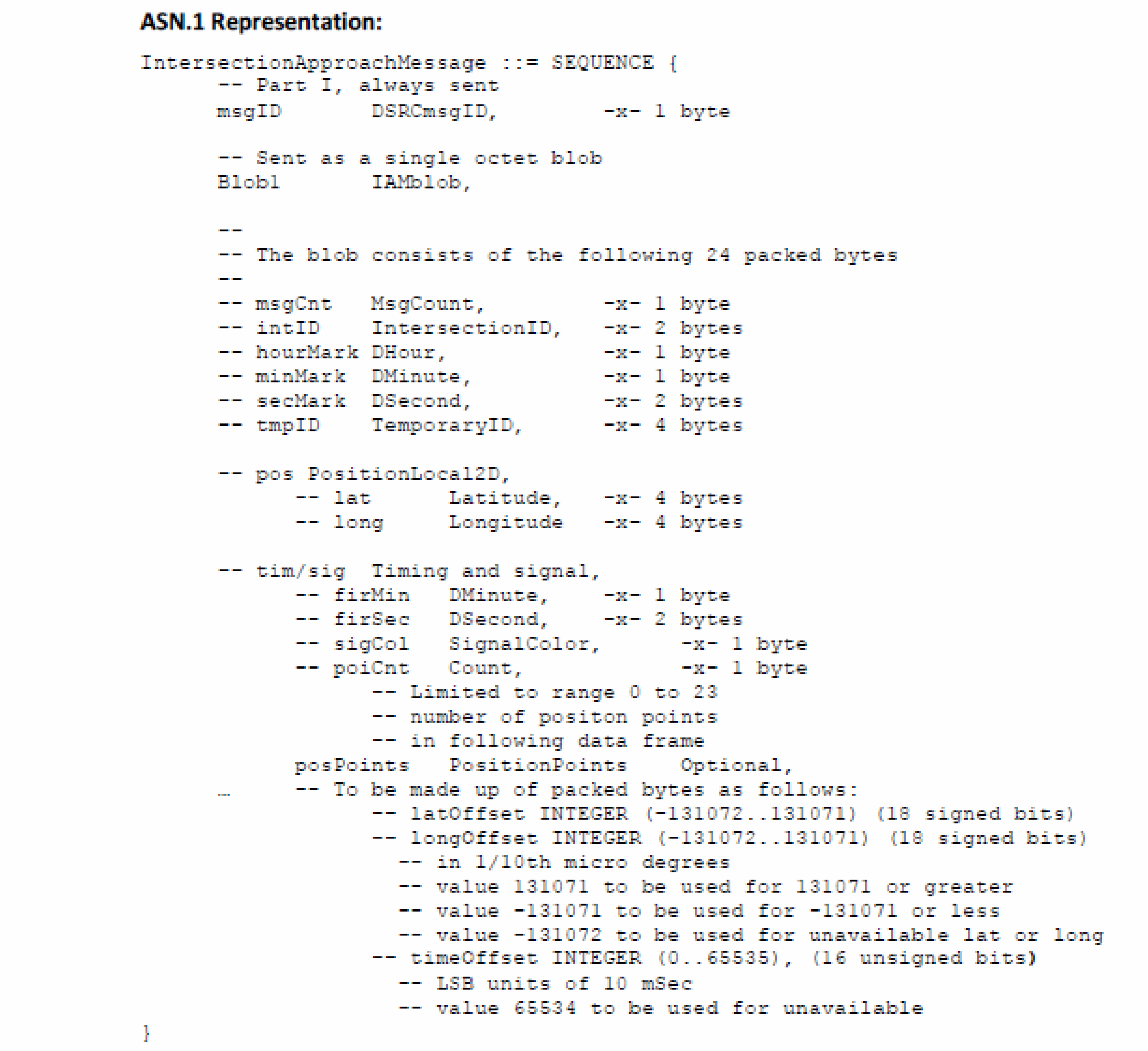
ASN.1 Representation:

```
ConnectedSafetyMessage:: = SEQUENCE {
msgID DSRCmsgID,   -- 1 byte

      --  Sent as a single octet blob
      blob1 CSMblob,
      --
      -- The blob consists of the following 26 packed bytes
      --
      -- msgCnt   MsgCount,          -x- 1 byte
      -- id       TemporaryID,       -x- 4 bytes
      -- hourMark DHour,             -x- 1 byte
      -- minMark  DMinute,           -x- 1 byte
      -- secMark  DSecond,           -x- 2 bytes

      -- pos      PositionLocal2D,
            -- lat      Latitude,    -x- 4 bytes
            -- long     Longitude,   -x- 4 bytes
            -- head     Heading,     -x- 2 bytes

      -- direct         Direction
            -- speed    Speed,       -x- 2 bytes
            -- laneNumb LaneNumber,  -x- 1 byte

      -- info     VehicleInfo
            -- length   VehicleLength,     -x- 2 bytes
            -- maxAcc   MaxAcceleration,   -x- 2 bytes
            -- maxDec   MacDeceleration,   -x- 2 bytes
            -- served   Served,            -x- 1 byte

      … -- # LOCAL_CONTENT
      }
```

Figure 3-6 Connected safety message ASN. 1 representation

3.2.3 Intersection Approach Message (IAM)

The IAM is populated by the traffic optimizer located at the intersection. Along with its message ID number, it contains the fixed intersection ID number followed by the time the message is sent and the vehicle ID number that the message is meant for. It then lists a set of trajectory points (in terms of 2D GPS coordinates) along with a time for each point for the autonomous vehicle to follow. In the message structure there is also a data element representing the color of the light for the lane that the connected vehicle is approaching. The order of each of the elements comprising the IAM can be seen in the ASN.1 representation in Figure 3-7.



ASN.1 Representation:

```
IntersectionApproachMessage ::= SEQUENCE {
      -- Part I, always sent
      msgID       DSRCmsgID,         -x- 1 byte

      -- Sent as a single octet blob
      Blob1       IAMblob,

      --
      -- The blob consists of the following 24 packed bytes
      --
      -- msgCnt   MsgCount,          -x- 1 byte
      -- intID    IntersectionID,    -x- 2 bytes
      -- hourMark DHour,             -x- 1 byte
      -- minMark  DMinute,           -x- 1 byte
      -- secMark  DSecond,           -x- 2 bytes
      -- tmpID    TemporaryID,       -x- 4 bytes

      -- pos PositionLocal2D,
            -- lat      Latitude,    -x- 4 bytes
            -- long     Longitude    -x- 4 bytes

      -- tim/sig  Timing and signal,
            -- firMin   DMinute,     -x- 1 byte
            -- firSec   DSecond,     -x- 2 bytes
            -- sigCol   SignalColor,       -x- 1 byte
            -- poiCnt   Count,             -x- 1 byte
                  -- Limited to range 0 to 23
                  -- number of positon points
                  -- in following data frame
            posPoints   PositionPoints     Optional,
      …     -- To be made up of packed bytes as follows:
                  -- latOffset INTEGER (-131072..131071) (18 signed bits)
                  -- longOffset INTEGER (-131072..131071) (18 signed bits)
                    -- in 1/10th micro degrees
                    -- value 131071 to be used for 131071 or greater
                    -- value -131071 to be used for -131071 or less
                    -- value -131072 to be used for unavailable lat or long
                  -- timeOffset INTEGER (0..65535), (16 unsigned bits)
                    -- LSB units of 10 mSec
                    -- value 65534 to be used for unavailable
}
```

Figure 3-7 Intersection approach message ASN.1 representation

3.3 Connected Vehicle Hardware

The hardware for the non-autonomous connected vehicle comprises a heavy duty pelican case (Figure 3-8, item 1) that houses the electronic equipment. The NUC (Figure 3-8, item 3) is a fully functional computer, running Ubuntu 15.04, which allows the user to load and change applications on the radio. It also displays the messages the MK5 (2) sends and receives in real time for quick verification of communication. The MK5 and NUC are powered by a 12V battery (not shown in Figure 3-8) which can be charged via an inlet mounted on the side of the case. Also mounted on the side of the case are antenna mounts (7) allowing communication to take place even if the lid to the suitcase is closed. Toggles switches (8) are used to control the power to the NUC, radio or battery. A voltmeter (9) is used to display the current voltage of the battery when it is powering the NUC and MK5.

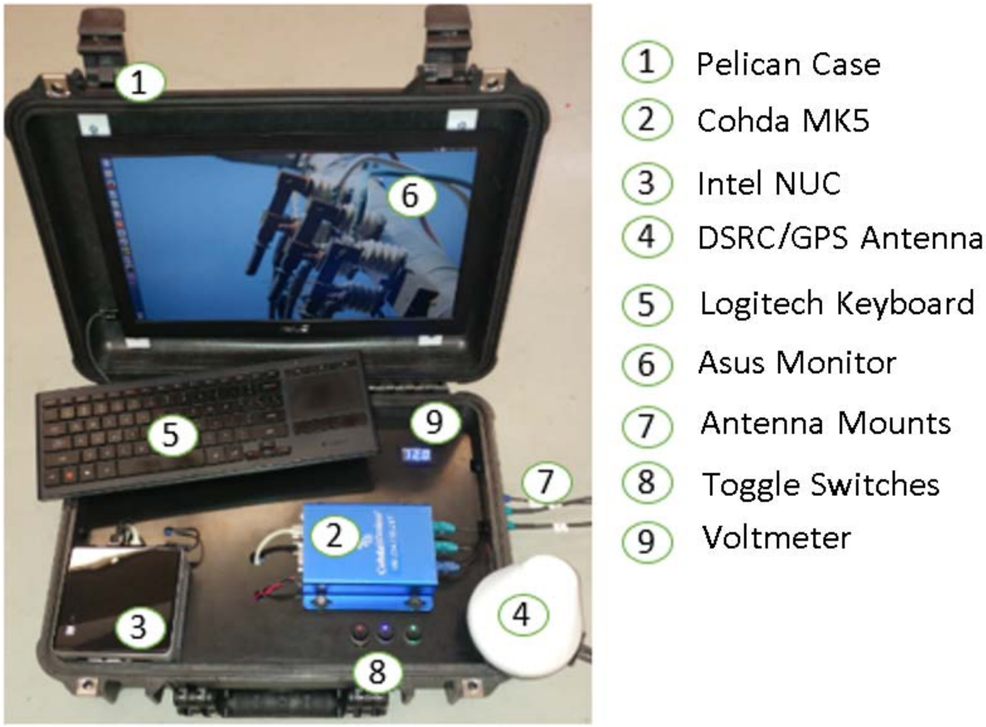


Figure 3-8 Connected vehicle suitcase

3.4 Autonomous Vehicle System

Figure 3-9 depicts the six components that comprise the autonomous system. The first five components existed prior to this project start, although several modifications were needed to update the system. The last item in the figure represents the new work that is being done so that the autonomous system can communicate with the traffic control system and execute appropriate commands.

The vehicle is a hybrid Toyota Highlander, shown in Figure 3-10. This system was originally developed for participation in the DARPA Urban Challenge competition. Actuators were added to control the steering and shifting. The brakes and accelerator are drive-by-wire systems and these were reversed engineered so that emulated signals are generated by UF computers and sent to the Toyota control computer.

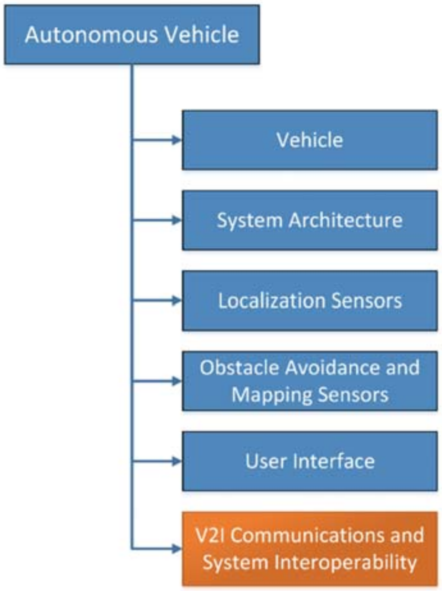

Autonomous Vehicle
Vehicle
System Architecture
Localization Sensors
Obstacle Avoidance and Mapping Sensors
User Interface
V2I Communications and System Interoperability

Figure 3-9 Components of autonomous vehicle

Figure 3-10 Autonomous vehicle

3.4.1 System Architecture

Figure 3-11 provides an overview of the system architecture. The lower blue box shows GPS and IMU sensors used for vehicle localization as well as vision and ladar sensors that are used for obstacle detection and terrain mapping.

The Intelligence Element box shows several Situation Assessment Specialist components that report findings. For example, one component continuously reports true or false as to whether the current travel lane is free of obstacles and obstructions for the next 20 meters. Behavior Specialists use the findings information to generate a score as to whether its associated behavior was appropriate at each instant. In the DARPA Urban Challenge competition, there were six possible vehicle behaviors and thus six Behavior Specialists. The six behaviors were: 1) Lane Following, 2) Change Lane, 3) Re-plan and Reverse Direction, 4) Parking Lot Maneuvering, 5) Parking, and 6) Intersection Behavior. In most instances, the Lane Following behavior is most appropriate. When the lane is detected as blocked, the Change Lane or Reverse Direction behaviors may be most appropriate. When in the vicinity of an intersection, the Intersection Behavior is scored highest and is selected.

The resulting selected behavior is implemented by the Smart Arbiter component which provides input to the Control Element, which ultimately controls the vehicle steering, throttle, and braking actuators.

3.4.2 Localization Sensors

Accurate position and orientation information are very important for autonomous navigation. Systems are available that can determine vehicle position to an RMS accuracy ranging from 2 cm to 3 m. Cost and convenience increase significantly as the position error is reduced. For this project, an accuracy of approximately 10 cm is desired. To achieve this, a typical GPS unit must operate in a differential mode where corrective signals are received from ground-based sensors at known points. Commercial vendors are providing this service at an annual cost.

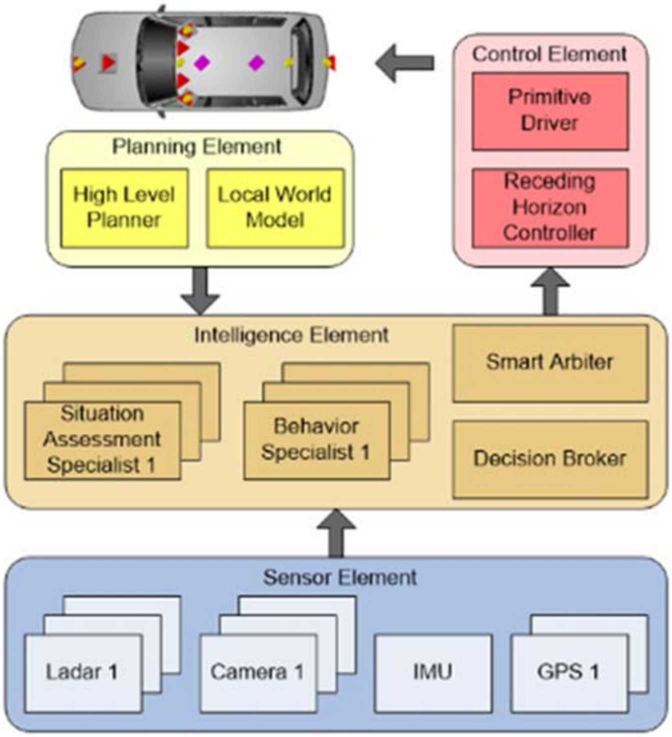


Figure 3-11 System architecture

A GPS combined with an Inertial Measurement Unit (IMU) can achieve accurate position and orientation data at high data rates. The addition of the IMU also allows for somewhat accurate positioning during times when the GPS signal is lost such as when going under a bridge. Typical error drift is on the order of 5 m when GPS is lost for 60 seconds. This can be accommodated by the autonomous vehicle since lane markings and terrain sensors can maintain the vehicle in its travel lane during these brief periods.

One update to the vehicle that is being done in this project is to replace the prior GPS system with a much lower cost system. A Novatel GPS was used during the DARPA Urban Challenge. This unit costs several thousands of dollars and requires an annual $1500 payment for subscription to corrective signals that are broadcast from base stations throughout the country. The new system developed uses a low-cost GPS sensor (~$125). GPS correction signals are received at no cost over the internet via the Florida DOT Florida Permanent Reference Network (FPRN). Centimeter level accuracy is expected from the final system.

3.4.3 Obstacle Avoidance and Mapping Sensors

In the DARPA Urban Challenge project, UF researchers were able to identify lane markings and obstacles using a combination of vision and ladar sensors. Six SickLMS 290 lidar sensors are mounted to the vehicle. Two additional long-range lidar sensors are mounted on the front quarter panels. These are augmented by two monocular vision sensors mounted at the top corners of the vehicle which map the location of lane markings. This information is particularly important during the lane following behavior.

3.4.4 User Interface

The development of an effective user interface is very important for this project. The interface must make it easy for the operator to set up and run a demonstration as well as be able to demonstrate to observers the abilities of each sensor type and how behaviors are decided upon by the vehicle. The in-vehicle user interface is being updated. The tablet that was used for this purpose is being replaced by a new computer and touch screen. This will allow the user to initialize the vehicle so it can be driven manually or placed in autonomous mode.

3.4.5 Hardware Modifications

The bulk of the work done on the autonomous vehicle so far has been to replace a custom hardware component that was responsible for the electrical interface between the autonomy and the Toyota Highlander Hybrid (THH). Figure 3-12 represents an additional component in the hardware interface that is responsible for steering and shifting. The hardware component measured vehicle sensor values related to the operation of the vehicle, i.e., the sensors of the brake and accelerator pedals, output sensor values to the ECUs to control braking and accelerating, and controlled various digital I/O for the vehicle (such as the turn signals).

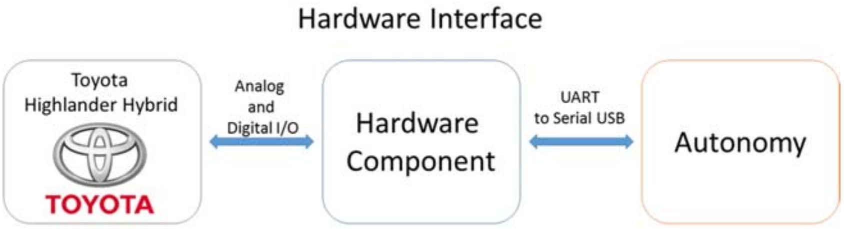


Figure 3-12 Vehicle-Autonomy hardware interface

Figure 3-13 National Instruments (NI) myRIO-1900 to replace the previous hardware

The hardware component was replaced with a NI myRIO-1900 which can be seen in **Error! Reference source not found.**. The myRIO was selected because it is a convenient OTS component that is easy to replace if it fails. Because of hardware differences between the original component and the myRIO, there were extensive changes that were needed to be made for it to be compatible with the THH. Signal Conditioners were needed to allow the myRIO to read digital voltages higher than allowable. Similarly, Opto-isolated relays were added to allow the myRIO to "output" voltages higher than allowable.

Additional hardware components that were replaced or added:

- 12 V batteries
- Cover plate for the center console
- Power cutoff switches
- Signal conditioners
- Various wiring.

Figure 3-14 to 15 show the configuration of the components on the vehicle. Figure 3-14 shows the completed replacement of the hardware component in the cockpit. Almost everything modified lies between the two front seats. Under the cover, solid state relays, OI relay SPDT, terminal block, regulator and conditioner are installed (Figure 3-15.) A detailed top view of the area between the two seats is shown in Figure 3-16. On the left, a top view of the cover plate with the box containing the NI myRIO device is shown. Figure (right) provides a view of the bottom of the box where the -16 3
.cables connect to the box. This connector is for all relevant sensors The functionality of the main components was discussed in previous sections. Along with the hardware changes, new code was created to give the myRIO the same capabilities as the original hardware component.

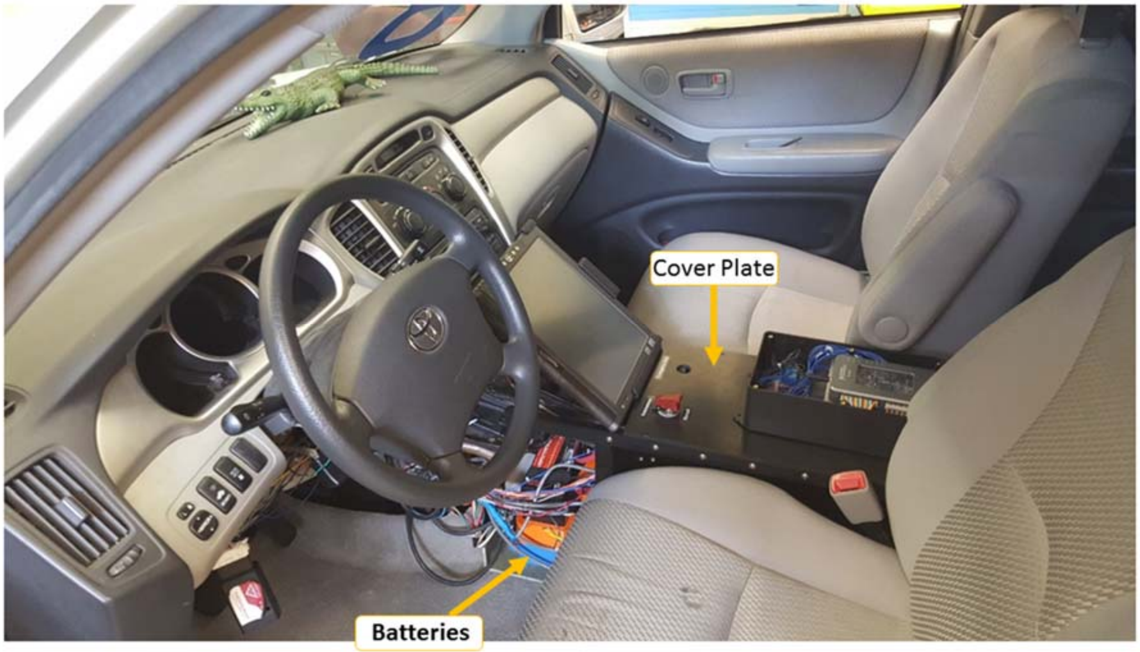


Figure 3-14 Completing the replacement of the hardware component

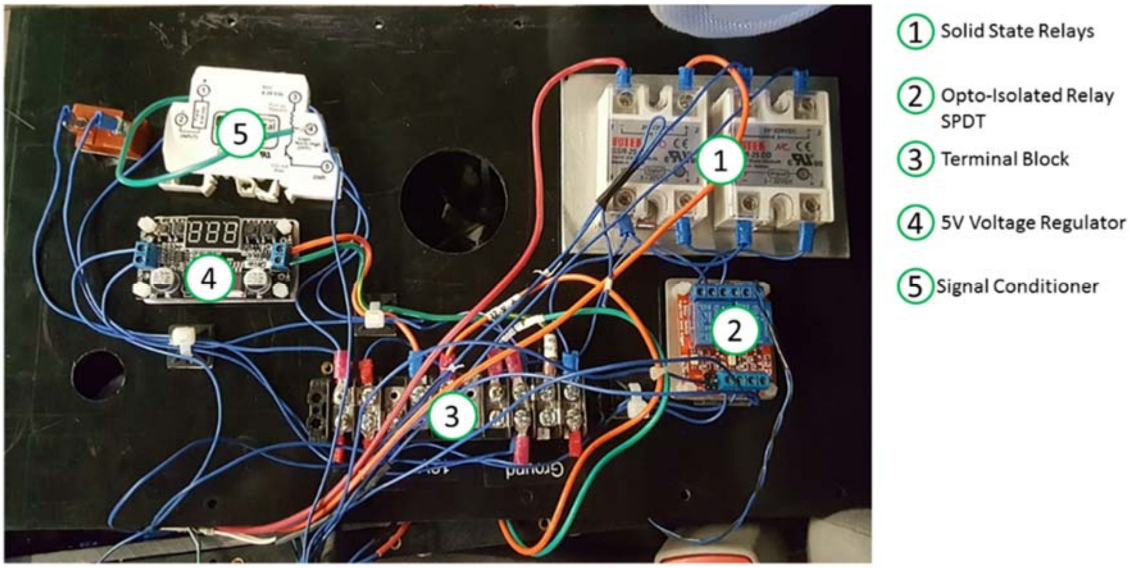


Figure 3-15 The underside of the cover plate

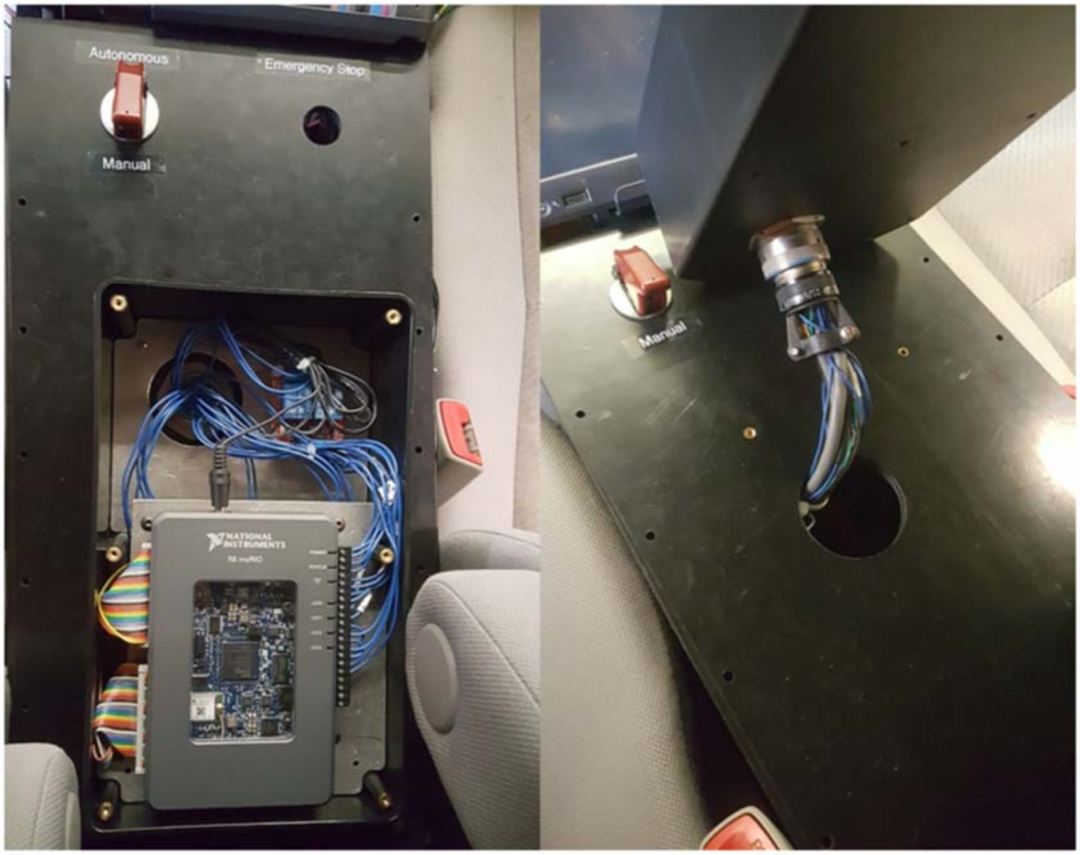


Figure 3-16 (Left) Top view of the cover plate. (Right) View of the bottom of the box

3.4.6 V2I Communications and System Interoperability

The original autonomy did not have the capability of perceiving a signalized intersection as anything more than a four-way stop. So a new behavior was added to allow the vehicle to react accordingly when approaching a signalized intersection. Along with the behavior, additional Situation Assessment Specialists and Behavior Specialist were created.

At least two Situation Assessment Specialists must be created: one to know the vehicle is approaching a signalized intersection and one to know if the vehicle has received a DSRC communication from the intersection. The new Behavior Specialist monitors these additional specialists and other previous ones to determine if it is appropriate to begin using the signalized intersection behavior. The signalized intersection behavior will be able to receive a trajectory sent from the intersection and use this trajectory to create the vehicle's path and desired speed.

4 Testing and Evaluation at TERL

The tests conducted at the TERL consisted of several human-driven vehicles, connected vehicles configured to emulate a level 2 autonomous vehicle, and vehicles equipped to emulate fully autonomous level 4 vehicles.

This chapter provides an overview of the testing, the testing scenarios, and summarizes our findings regarding the overall performance of the signal control optimization process, the trajectory optimization, and the equipment performance, including sensor fusion process and functionality.

4.1 Overview of Testing Effort

The main testing effort took place on Thursday, May 25th and Friday, May 26th, 2017. Additional testing took place on October 19th and 20th, 2017, to further refine the functionality of the autonomous vehicle, which malfunctioned during the initial testing (faulty brakes). The four-leg intersection at the TERL (Figure 4-1) has six approaching lanes and four departing lanes and is located on level terrain. Turning lanes were not be utilized throughout the testing to simplify the phasing patterns. There was no lane-changing occurring once a vehicle arrival is detected and no pedestrians were assumed to be present in the vicinity of the intersection. A total of six vehicles were used

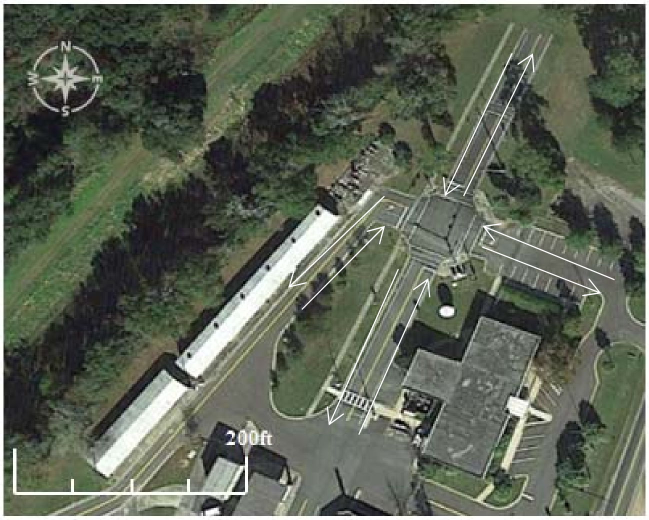
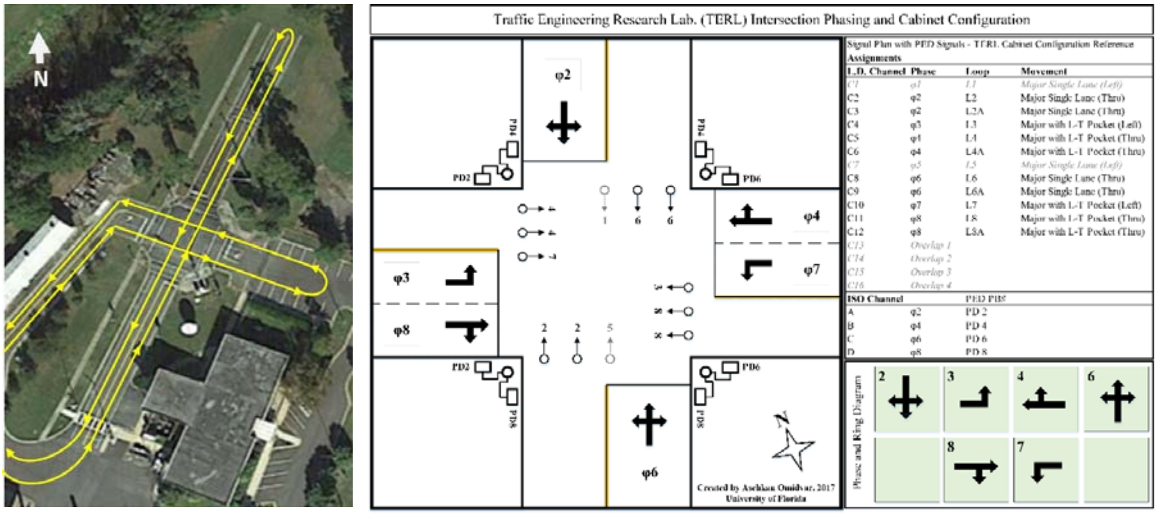


Figure 4-1 TERL intersection

Vehicle trajectories were monitored using a GPS system in each vehicle, a Cohda MK5 radio to transmit and receive DSRC messages in each connected vehicle, localization sensors in the autonomous vehicle, one radar, and two cameras at the intersection. The minimum communication distance was assumed to be 150 ft. A drone was used to record video of the testing (see http://avian.essie.ufl.edu/gallery/).

4.2 Test Scenarios

Figure 4-2 shows an aerial photo of the TERL and the test signalized intersection. Scenarios vary in the types of vehicles used (conventional, CV, AV) and vehicle inter-arrival times (i.e., level of demand.) Each testing scenario includes approximately 15 minutes of set up time and 15 minutes of a continuous testing run with vehicles traveling along a predetermined path. The maximum speed for all scenarios and vehicles is 15 mph. The scenarios start from as simple as 2 CVs, up to a mixed traffic of 4 AV and 2 conventional vehicles.


N
Traffic Engineering Research Lab. (TERL) Intersection Phasing and Cabinet Configuration
φ2
PD4
PD2
PD6
φ4
φ7
φ3
φ8
PD2
PD8
PD6
PD8
φ6
1
6
6
2
2
5
Created by Aschkan Omidvar, 2017
University of Florida
Phase and Ring Diagram
2
3
4
6
8
7

Signal Plan with PED Signals - TERL Cabinet Configuration Reference

Assignments

L.D. Channel	Phase	Loop	Movement
C1	φ1	L1	Major Single Lane (Left)
C2	φ2	L2	Major Single Lane (Thru)
C3	φ2	L2A	Major Single Lane (Thru)
C4	φ3	L3	Major with L-T Pocket (Left)
C5	φ4	L4	Major with L-T Pocket (Thru)
C6	φ4	L4A	Major with L-T Pocket (Thru)
C7	φ5	L5	Major Single Lane (Left)
C8	φ6	L6	Major Single Lane (Thru)
C9	φ6	L6A	Major Single Lane (Thru)
C10	φ7	L7	Major with L-T Pocket (Left)
C11	φ8	L8	Major with L-T Pocket (Thru)
C12	φ8	L8A	Major with L-T Pocket (Thru)
C13	Overlap 1		
C14	Overlap 2		
C15	Overlap 3		
C16	Overlap 4		

ISO Channel		PED PBS
A	φ2	PD 2
B	φ4	PD 4
C	φ6	PD 6
D	φ8	PD 8

Figure 4-2 TERL intersection, testing route, and phasing

During the initial TERL tests, we used 4 simulated autonomy vehicles. Cars instrumented with DSRC acted as AVs by having a passenger using an application with a user-friendly GUI that received simplified trajectory messages and displayed recommended speeds for the driver for each approach.

To evaluate the tested scenarios and the performance of vehicles and effectiveness of tested scenarios, several measures are tracked and analyzed. Data were recorded and stored on-site after each scenario from each vehicle GPS system, the traffic signal controller, the radar, the communications components, intersection cameras, and a drone. The GPS systems measure the actual trajectory of each vehicle. We assessed how well the AVs/CVs were able to follow the recommended trajectories. In addition, we evaluated the performance of our car-following model within the system by comparing the estimated trajectories with the actual trajectories based on data obtained from a traffic radar. Therefore, the performance measures for operations were:

- Mean Travel Time (vehicle information from GPS, DSRC suitcases and intersection cameras)
- Mean Travel Delay (vehicle information from GPS, DSRC suitcases and intersection cameras)
- Phasing and Green Duration (signal timing data from algorithm, intersection cameras)
- Vehicle Trajectories (actual vs. recommended trajectories by vehicle type)

Moreover, we collected data for each component separately and assessed the performance of each one. In detail, we analyzed run-time, solution quality and recommended SPaT and trajectories to evaluate the entire system, the communication latency and accuracy, the signal controller and cabinet along with signalization interface and setup, etc. Interested readers may refer to the project webpage at ‘avian.essie.ufl.edu’ for additional information regarding equipment and for videos of the testing.

4.3 Performance Measurements – Hardware And Software

The performance measures used to evaluate hardware and software include algorithm performance and runtime, DSRC communications reliability, vehicle operations reliability, algorithm-signal controller communications reliability, and radar sensing reliability.

Algorithm outputs captured runtime, recommended trajectories, and recommended signal timings. These were used to evaluate the reliability and performance of the algorithm.

Communications reliability includes transmission times, errors in data transmissions, and potential malfunctions of the communications components. The reliability of communication between the algorithm and the signal controller was also evaluated.

The performance of the radar was evaluated by comparing the actual arrivals to those detected by the radar. Delay in recognition and fusion of the data from the radar and the DSRC was obtained.

Equipment malfunctions or communication malfunctions during a scenario were noted in the scenario log. Table 4-1 summarizes the hardware and software – related performance measures obtained during the testing.

It was determined that the MATLAB code runs instantaneously in simulation mode. However, this only includes the optimization computation time and not the communication time. Under real-time mode, the running time goes up to 0.1 sec per message to be processed. This is due to the addition of fusion, preprocessing and post processing components to the system.

Table 4-1 Summary of performance measurements for hardware and software

Performance Measure	Measurement Method
Algorithm Performance	Processing time, recommended signal timings and trajectories.
Algorithm to Vehicle Communication Malfunctions	Communication time between algorithm and vehicle. Differences in trajectories between recommended and actual (measured through GPS, intersection cameras.) Vehicle types will be distinguished (connected, autonomous).
Vehicle Malfunction	Observation of vehicle operations through GPS and video.
Signal Communication Malfunction	Signal controller display observed by video, algorithm optimization results.
radar Fusion Malfunction	Discrepancies between vehicle arrivals and vehicle recognition by the radar. Discrepancies in the data fusion component.
Total Communication Malfunction	Actual vehicle trajectories, signal control.

4.4 Trajectory Optimization

The goal of the trajectory optimization component is to assign trajectories to automated vehicles that allow for the most efficient flow of traffic through the intersection. The optimization must also account for the presence of connected and conventional vehicles, which are human-driven, and thus their exact behavior cannot be predicted. During the TERL tests, we used simulated autonomy; essentially, cars instrumented with DSRC acted as automated vehicles by having a passenger use an application that received trajectory messages and displayed recommended speeds for the driver. All references to "automated" vehicles therefore refer to simulated automated vehicles. In our testing, we evaluated how well the automated and connected vehicles were able to follow the recommended trajectories. Additionally, we assessed the performance of the car-following model used in our optimization to predict the behavior of conventional and connected vehicles. This assessment was conducted by comparing our estimated trajectories with the actual trajectories observed in the field based on data obtained from a traffic radar. The metric we use to assess how well a vehicle followed a recommended trajectory is the root mean square error (RMSE). This is a measure of the absolute distance in feet between where the vehicle is at some point in time and where it is supposed to be at that same instant based on estimated trajectory.

We tracked all automated and connected vehicles using the GPS and speed data received from the DSRC communication and the radar. These data can be aligned and compared with the trajectories computed by the optimization component of the algorithm. Table 4-22 shows the RMSE between the expected and actual trajectories of automated and connected vehicles averaged over a sample of 23 trajectories. The min RMSE is the smallest RMSE observed at any point during the trajectory and the max RMSE is the largest observed. We consider uncertainty of one standard deviation (95% interval).

Table 4-2 Root Mean Square Error (RMSE) for assigned vs. actual trajectories

Min RMSE (ft.)	Max RMSE (ft.)	Mean RMSE (ft.)
4.333 ± 3.507	64.856 ± 42.614	33.130 ± 18.587

It was observed that the trajectories computed by the optimization tended to be much more nonlinear than the observed trajectories (Figure 4-3). During our experiments we presented a recommended speed to the drivers of our automated vehicles with a graphical user interface (GUI) to have them attempt to follow the trajectory as closely as possible. However, this does not allow for them to account for nonlinearities in the optimized trajectory. Further, the RMSE could be reduced by improving the time synchronization of our real-time system. With a clock synchronized with the DSRC communication, the system is able to account for the 10-20 feet that a vehicle will move during the time it takes for a message to be sent via DSRC from the vehicle to the intersection, and then to receive the trajectory.

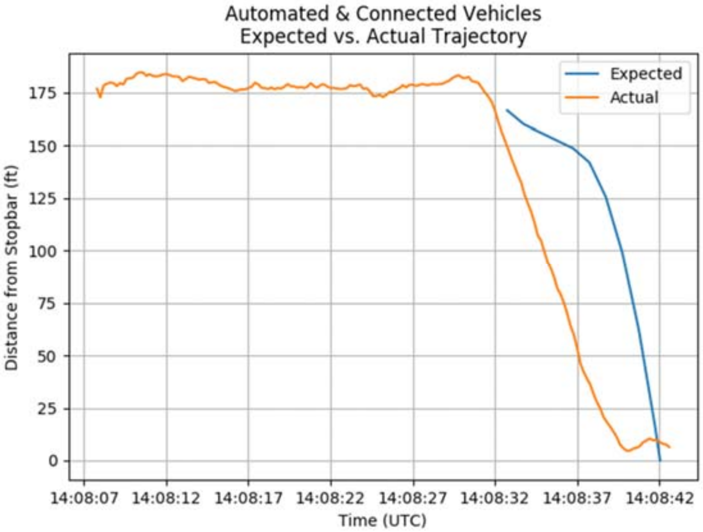


Figure 4-3 Comparison of actual vs. assigned trajectory for an automated/connected vehicle

4.5 Conventional Vehicles

During our testing, in almost all instances when a conventional vehicle approached the intersection, they were the lead vehicle in their lane. Depending on the prevailing signalization, we observed that our algorithm estimated two different types of trajectories. If the lane had the right-of-way (green signal) when the conventional vehicle arrived, the estimated trajectory would be linear. Otherwise, we estimated a nonlinear trajectory that would attempt to pass through the intersection at the start of the green time for their lane. Table 4-3 shows the RMSE for the expected vs. actual trajectories averaged over a sample size of 12 conventional vehicle trajectories. We observed that our conventional vehicles maintained their speed consistently through the entire test (Figure 4-4); however, the car-following model occasionally predicted a nonlinear trajectory which did not match the actual trajectory (Figure 4-4, right side). Due to this discrepancy, we occasionally observed a large RMSE between the expected and actual trajectories (max RMSE in Table 4-3).

Table 4-3 RMSE for expected vs actual trajectories for conventional vehicles

Min RMSE (ft.)	Max RMSE (ft.)	Mean RMSE (ft.)
0.061 ± 0	75.499 ± 53.113	32.054 ± 20.923

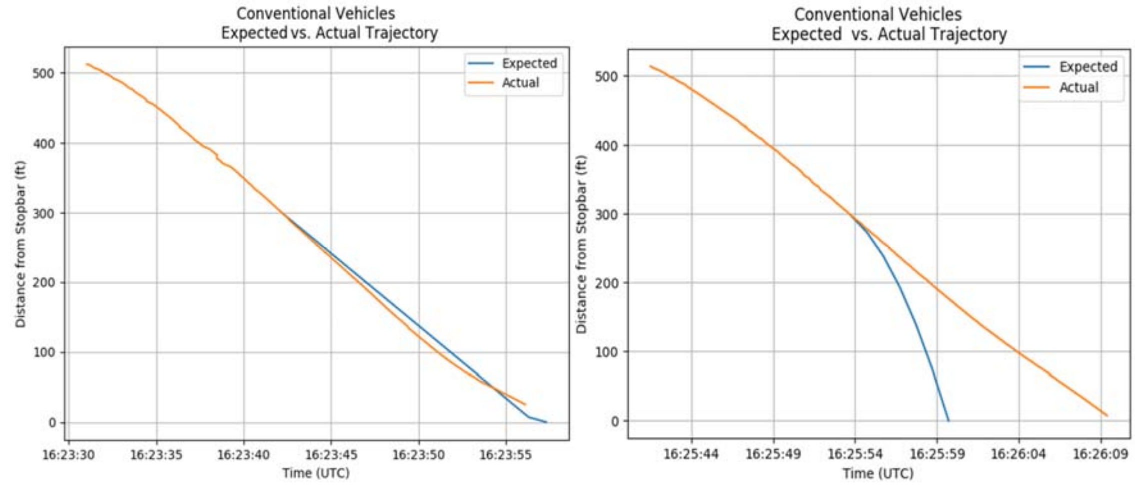


Figure 4-4 Comparisons of expected vs. actual trajectories for conventional vehicles

4.6 Overall Performance of the Optimization Algorithm

In order to quantify the performance of the optimization algorithm, we computed the travel time delay for each vehicle as the time difference between the base travel time (theoretical minimum travel time based on prevailing free flow speed) and the computed travel time. The base travel time is defined as the time required for the vehicle to travel through the intersection if there is no traffic signal. The computed trajectories provide the theoretical travel time of each vehicle from detection to departure from the intersection stop bar.

Table 4-4 indicates the minimum, average, and maximum of travel time delays for all vehicles that traveled through each of the four intersection approaches while testing at the TERL. For all approaches, the average delay is close to the minimum, confirming the intent of the trajectory optimization. The NB approach—which was equipped with the radar—shows the least average delay as more observations with lower delays are included in this category.

Table 4-4 Travel time delay per approach estimated by the optimization algorithm

Approach	Min Delay	Average Delay	Max Delay	Count
SB	0.0	2.2	6.0	16
WB	0.2	2.2	7.7	6
NB	0.0	0.5	3.8	20
EB	0.0	1.6	3.8	8

4.7 Sensor Fusion

The goal of the proposed sensor fusion system is to fuse localization information from multiple sources to optimally estimate the position and speed of every vehicle approaching the intersection. This fused estimate can then be provided as inputs for the optimization. For such an application, it is also important to classify every fused track as either originating from a conventional or a DSRC-equipped vehicle. To do this, several challenges inherent to multi-sensor multi-target tracking systems must be addressed. Namely, this involves temporally aligning the incoming data, associating tracks between sensors, and fusing the associated tracks. After describing our sensor fusion framework, we explain our approach to solve each of these specific tasks.

4.7.1 Sensing Modalities

We consider a scenario where zero or more vehicles approaching an intersection are equipped with DSRC transceivers (OBUs). Accordingly, there is an RSU present at the intersection set up to receive BSMs transmitted from the OBUs on the 5.9 GHz band dedicated for ITS applications. The OBUs are designed to transmit a BSM at some fixed interval, e.g., every 50 or 100 ms. The accuracy and precision of the localization information contained in the BSM is dependent on the quality of the GPS available to the OBU. We use Wide Area Augmentation System (WAAS) GPS, which reduces the uncertainty in uncorrected GPS to about 2-3 meters in latitude and longitude. Tests were conducted to roughly characterize the noise in the reported latitude and longitude by our OBUs. The GPS in our OBUs update at a rate of 5 Hz, or once every 200 ms. The observation $\mathrm{o}(k)$ of a vehicle's state at time k is obtained from a BSM as:

$$\mathbf{o}(\boldsymbol{k}) = [\mathbf{Lat}(\boldsymbol{k})\ \mathbf{Lon}(\boldsymbol{k})\ \boldsymbol{\theta}(\boldsymbol{k})\ \boldsymbol{v}(\boldsymbol{k})]^{\mathbf{T}} \qquad \text{Equation 7}$$

In Equation 7, $\theta(k)$ is the heading of the vehicle in degrees counter-clockwise from true north and $v(k)$ is the speed of the vehicle.

In mixed traffic situations, many vehicles approaching an intersection will be conventional vehicles without any DSRC capabilities. One way to still successfully surveil all traffic surrounding the intersection is to use an advanced detection Doppler traffic radar. Advanced detection radars can track oncoming vehicles over 200 meters away, obtaining accurate range and speed information. We observed that at distances greater than 100 meters, the radar has significant difficulty identifying the lateral position of a vehicle within a lane. Hence, we only make use of the range and speed information reported by the radar. The observation $\mathrm{o}(k)$ obtained from the radar for each vehicle is:

$$\mathbf{o}(\boldsymbol{k}) = [\ \boldsymbol{y_r}(\boldsymbol{k})\ \dot{\boldsymbol{y}}_{\boldsymbol{r}}(\boldsymbol{k})\ ]^{\mathbf{T}} \qquad \text{Equation 8}$$

where $y_r(k)$ and $\dot{y}_r(k)$ are the range and speed information respectively of the approaching vehicle within a 2D Cartesian coordinate frame with the radar at the origin. The detections we receive already account and correct for the mounting height of the radar. For the sake of simplicity, we assume a single Doppler radar is mounted on a mast arm directly facing oncoming traffic. Radars track vehicles based on reflections from a corner or side of the vehicle; however, as this reflection point is unknown and can vary, it is difficult to account for it when fusing position information obtained from DSRC. This is because the DSRC may be reporting localization information from a specific point on the vehicle, e.g., the center of the front axle of the vehicle. In our system, information for all vehicles in the field-of-view of the radar is sent to the sensor fusion system at a rate of 20 Hz.

4.7.2 Tracking Models

In order to fuse the information from the radar with BSMs received over DSRC, we estimate the uncertainty in each sensor's measurements with a linear Gaussian Kalman Filter (KF). Essentially, the system maintains tracks at the sensor level and then sends these tracks a central processing unit to time-align and fuse them. We carry out tracking in a global coordinate frame with the Universal Transverse Mercator (UTM) system; this allows us to convert GPS coordinates to a system that is locally Cartesian and uses interpretable units, e.g., meters. UTM preserves the shape of small areas on a map and the grid coordinates permit easy trigonometric. We model the dynamics of all vehicles using a constant velocity motion model with vehicle acceleration captured as noise. The motion model used for DSRC encodes the vehicle state $\mathrm{x}(k)$ at time k as:

$$\mathbf{x}(\boldsymbol{k}) = [\,\boldsymbol{x}(\boldsymbol{k})\;\boldsymbol{y}(\boldsymbol{k})\;\dot{\boldsymbol{x}}(\boldsymbol{k})\;\dot{\boldsymbol{y}}(\boldsymbol{k})]^{\mathbf{T}} \qquad \text{Equation 9}$$

Here, $(x(k),\, y(k))$ is the position of the vehicle in UTM Easting and Northing, converted directly from the raw GPS reported in the BSM. Likewise, $\dot{x}(k)$ and $\dot{y}(k)$ are speeds in the UTM Easting and Northing directions. We compute Equation 9 from Equation 7 as follows, omitting the time parameter k for notational simplicity:

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{y}} \end{bmatrix} = \begin{bmatrix} \mathbf{deg2UTM(Lat,\; Lon)} \\ \mathbf{deg2UTM(Lat,\; Lon)} \\ \boldsymbol{v}\cos\boldsymbol{\theta} \\ \boldsymbol{v}\sin\boldsymbol{\theta} \end{bmatrix} \qquad \text{Equation 10}$$

We use a standard conversion of WGS84 GPS coordinates to UTM, represented in Equation 10 by $\mathrm{deg2UTM}$.

For the radar tracks, the state at time k is given by:

$$\mathbf{x}(\boldsymbol{k}) = [\,\boldsymbol{y}(\boldsymbol{k})\;\dot{\boldsymbol{y}}\;(\boldsymbol{k})\,]^{\mathbf{T}} \qquad \text{Equation 11}$$

and is computed from Equation 8 by:

$$\begin{bmatrix} \boldsymbol{y} \\ \dot{\boldsymbol{y}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{y}_{\mathbf{0}} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \cos\boldsymbol{\theta}' & -\sin\boldsymbol{\theta}' \\ \sin\boldsymbol{\theta}' & \cos\boldsymbol{\theta}' \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_{\boldsymbol{r}} \\ \dot{\boldsymbol{y}}_{\boldsymbol{r}} \end{bmatrix} \qquad \text{Equation 12}$$

where θ' is the orientation of the radar in degrees counter-clockwise from true north, and y_0 is the UTM Northing coordinate of the radar.

Track initiation, confirmation, and deletion is managed by a finite state machine. We empirically determine the number of consecutive observed and missed detections that control state transitions to prevent prematurely adding or dropping tracks. Tracks are considered confirmed after a sensor receives consecutive detections for approximately 1 second. Once confirmed, our system attempts to associate the track with the other sensor's set of confirmed tracks. Upon successful association, a new KF is created that takes as input the two associated tracks from each sensor and produces a final fused state estimate.

4.7.3 Track Alignment

When fusing information from multiple sensors, it is convenient to time-align the incoming data. For example, in our scenario, the WAAS GPS data is updated at 5 Hz whereas we receive radar data at 20 Hz. To handle this, we down-sample the radar data to 5 Hz and update all KFs at 5 Hz. Since we receive data asynchronously from each sensor, it is not guaranteed that the sensor timestamps from the radar and DSRC align; generally, this can be handled by linearly interpolating the data from each sensor to obtain estimates at fixed time intervals. In the next iteration of our system, we plan to up-sample the GPS data received via DSRC to carry out tracking at a faster rate, e.g., at 10 or 20 Hz.

4.7.4 Track-to-Track Association

When there are only two sensors, the problem of track-to-track association (T2TA) can be phrased as follows: "Given a track from sensor A, find a track from sensor B that isn't already associated with any other tracks from sensor A and that *most likely* originated from the same object as the track from sensor A, subject to the constraint that the corresponding association likelihood is greater than a threshold". Concretely, let S_1 and S_2 be the two sensors, each with a list of tracks at time t represented by state estimates $\hat{x}_i^1$ and $\hat{x}_j^2$ respectively, $i = 1 \mathinner{..} N_1$ and $j = 1 \mathinner{..} N_2$. Here, N_1 and N_2 are the number of tracks in each sensor's track list. We omit the time parameter t for notational simplicity. Likewise, the state covariances estimated by the KFs are P_i^1 and P_j^2. One can naturally frame the problem as a hypothesis test, where the null hypothesis H_0 is $\hat{x}_i^1 = \hat{x}_j^2$ for track i from S_1 and track j from S_2. Since our state estimates are produced by linear Gaussian KFs, we can define the following test-statistic T for a Chi-squared Hypothesis test:

$$\hat{\boldsymbol{\Delta}} = \left(\hat{\boldsymbol{x}}_i^1 - \hat{\boldsymbol{x}}_j^2\right) \qquad \text{Equation 13}$$

$$\boldsymbol{T} = \hat{\boldsymbol{\Delta}}^{\mathbf{T}} \mathbf{P}_{\hat{\boldsymbol{\Delta}}}^{-1} \hat{\boldsymbol{\Delta}} \leq \boldsymbol{\alpha} \qquad \text{Equation 14}$$

Where $\hat{\Delta}$ is zero-mean and $P_{\hat{\boldsymbol{\Delta}}}$ is the covariance matrix of $\hat{\Delta}$, defined as $P_i^1 + P_j^2 - P_{ij}^{12} - P_{ij}^{21}$ (Bar-Shalom & Huimin Chen, 2004). The final two terms are the cross-correlation terms for the two-tracks between the two sensors. These are needed to account for the fact the state estimates of the two-tracks in question are statistically correlated under the null hypothesis that the two-tracks in question originate from the same object. Intuitively, the test statistic can be interpreted as the distance between the two track state estimates, scaled by the amount of uncertainty in each state dimension.

The number of degrees of freedom for the Chi-squared hypothesis test is D, the dimension of $\hat{\Delta}$. Theoretically, one can select the null acceptance threshold α to correspond to a 95% or 99% confidence threshold. In practice, due to errors in time alignment and undetected sensor biases, one may have to experimentally increase α to find an acceptable threshold.

In our system, given a newly confirmed track i from sensor S_1, the test statistic is computed for all pairings of i with all tracks $j = 1 \ldots N_2$ from sensor S_2, also known as global nearest neighbors. If more than one track falls within the acceptance threshold for association, we choose the track with the smallest *T*. When a DSRC track from a connected vehicle is matched with a radar track, we can classify that radar track as originating from a connected vehicle, as opposed to a conventional vehicle.

4.7.5 Track-to-Track Fusion

We employ the Covariance Intersection (CI) algorithm (Julier & Uhlmann, 1997) for fusing the DSRC and radar observations for two associated tracks. The CI algorithm is an attractive solution since it computes a simple convex combination of the track means and covariances from multiple sensors. The optimal combination parameters can be computed using a freely-available optimization package, such as *scipy.minimize*. We use the default optimizer for *scipy.minimize*, SLSQP, and select the combination parameters that minimize the determinant of the inverse of the combined covariances.

4.7.6 Experimental Results at TERL

At the TERL tests, we had at most six vehicles navigating the intersection at any given time; four were instrumented with DSRC, and two were conventional. Our main objectives were to (1) correctly classify each approaching vehicle as either connected, automated, or conventional and (2) estimate their position and speed. We achieved 100% classification accuracy, which means that in every test run, a conventional vehicle was never mistaken for an automated vehicle (easy to do) and an automated vehicle was never mistaken for a conventional vehicle (much harder, since we have both radar and DSRC information from this vehicle to fuse). Figure 4-5 summarizes the results of the evaluation. This type of a figure is called a confusion matrix. The color gradient on the right represents the number of vehicles. Each square represents the fraction of arrivals that were classified as automated/connected or conventional with respect to the true vehicle types. For example, when the vehicle was conventional, our system never classified it as automated/connected (bottom left square).

We did not have access to ground-truth for evaluating the accuracy and precision regarding position and speed of each vehicle. Hence, we simply provide an example (Figure 4-6) of how the DSRC and radar are fused to produce an accurate estimate of the true position (and speed) of each vehicle. The green line shows the fused position estimate produced by the algorithm. Based on prior experimentation where ground-truth was available, we know that the radar produces more accurate estimates of the position; hence, the sensor fusion system "trusts" the data coming from the radar more, and the fused estimate weights the radar's estimate more heavily.

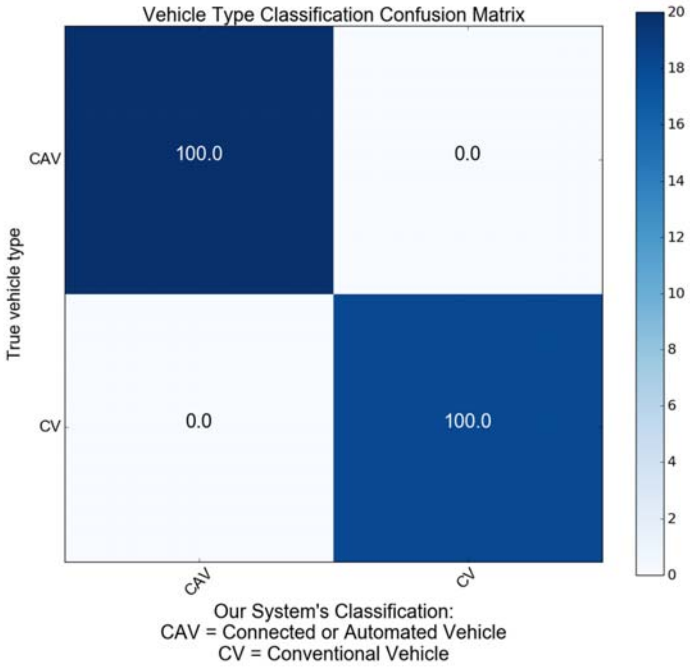


Figure 4-5 Classification results for the sensor fusion (number in the box represents percent accuracy)

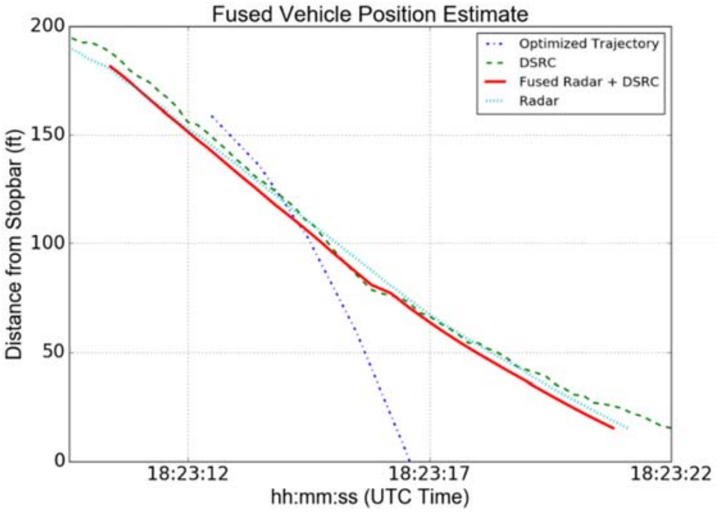


Figure 4-6 DSRC, radar, and fused radar and DSRC vehicle trajectories for a single vehicle driving towards an intersection

5 Conclusions and Recommendations

Significant improvements in automated and connected vehicle technologies are expected to create a revolution in how we move and move things. Automated vehicles can operate using a variety of sensors such as GPS, lidar, radar, and smart cameras, as well as terrain information, and they have the ability to communicate with infrastructure as well as surrounding vehicles. It is highly likely that in the not too distant future connected and autonomous will be operating side by side in large numbers, along with conventional vehicles. The objectives of this research were to develop, test, and deploy an intelligent real-time intersection traffic control system, designed to optimize simultaneously signal control and automated vehicle trajectories.

The system developed was first simulated in MATLAB. A total of 3000 scenarios were tested to consider varying demand levels, communication ranges, automated vehicle percentage, and saturation headways. The results showed that for lower saturation headways, the average travel time decreases as the automated vehicle percentage increases, since these vehicles can more easily follow shorter headways. On average, lower effective greens are allocated to higher demand levels. In these cases, there are more requests from conflicting movements to switch the right of way as the demand increases, while the signal can be extended without any interruption for the low demand scenarios. The same pattern can be seen for varying communication ranges: the higher the communication range, the better the ability to design a platoon of trajectories ahead of time, which can assign longer green intervals with fewer interruptions.

The system was then implemented at the TERL facility, and included various hardware and software components developed for the project (a local server, DSRC receiver for the server, interface to the signal controller, sensor fusion system, radio communication software, and hardware for vehicle to infrastructure communications.) Once developed, the system was tested under various demand and communications scenarios. The outputs and video footage (http://avian.essie.ufl.edu/gallery/) showed that the system is capable of providing optimal trajectories to automated vehicles in order to reduce delays.

Future work should expand the algorithm to consider congested conditions, lane changing, and the presence of pedestrians and bicycles.

References

Ahmane, M., Abbas-Turki, A., Perronnet, F., Wu, J., El Moudni, A., Buisson, J., & Zeo, R. (2013). Modeling and controlling an isolated urban intersection based on cooperative vehicles. *Transportation Research Part C: Emerging Technologies*, *28*, 44–62. https://doi.org/10.1016/j.trc.2012.11.004

Au, T.-C., Shahidi, N., & Stone, P. (2011). Enforcing Liveness in Autonomous Traffic Management. In *Proceedings of the 25th AAAI Conference of Artificial Intelligence* (pp. 1317–1322).

Bar-Shalom, Y., & Huimin Chen. (2004). Multisensor track-to-track association for tracks with dependent errors. In *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)* (p. 2674–2679 Vol.3). IEEE. https://doi.org/10.1109/CDC.2004.1428864

Dresner, K., & Stone, P. (2008). A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research*, *31*, 591–656. https://doi.org/10.1613/jair.2502

Gipps, P. G. (1981). A behavioural car-following model for computer simulation. *Transportation Research Part B*, *15*(2), 105–111. https://doi.org/10.1016/0191-2615(81)90037-0

Goodall, N., Smith, B., & Park, B. (2013). Traffic Signal Control with Connected Vehicles. *Transportation Research Record: Journal of the Transportation Research Board*, *2381*(2), 65–72. https://doi.org/10.3141/2381-08

Guler, S. I., Menendez, M., & Meier, L. (2014). Using connected vehicle technology to improve efficiency of intersections. *Transportation Research Part C: Emerging Technologies*, *46*, 121–131. Retrieved from http://dx.doi.org/10.1016/j.trc.2014.05.008

Julier, S. J., & Uhlmann, J. K. (1997). A non-divergent estimation algorithm in the presence of unknown correlations. In *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)* (pp. 2369–2373 vol.4). IEEE. https://doi.org/10.1109/ACC.1997.609105

Lee, J., Park, B. (Brian), & Yun, I. (2013). Cumulative Travel-Time Responsive Real-Time Intersection Control Algorithm in the Connected Vehicle Environment. *Journal of Transportation Engineering*, *139*(10), 1020–1029. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000587

Li, Z., Chitturi, M., Zheng, D., Bill, A., & Noyce, D. (2013). Modeling Reservation-Based Autonomous Intersection Control in VISSIM. *Transportation Research Record: Journal of the Transportation Research Board*, *2381*, 81–90. https://doi.org/10.3141/2381-10

Li, Z., Elefteriadou, L., & Ranka, S. (2014). Signal control optimization for automated vehicles at isolated signalized intersections. *Transportation Research Part C: Emerging Technologies*, *49*, 1–18. https://doi.org/10.1016/j.trc.2014.10.001

National Transportation Operations Coalition (NTOC), 2012. National Traffic Signal Report Card. NTOC, Washington, D.C.

Tachet, R., Santi, P., Sobolevsky, S., Reyes-Castro, L. I., Frazzoli, E., Helbing, D., & Ratti, C. (2016). Revisiting Street Intersections Using Slot-Based Systems. *Plos One*, *11*(3), e0149607. https://doi.org/10.1371/journal.pone.0149607

Wu, J., Abbas-Turki, A., & El Moudni, A. (2012). Cooperative driving: An ant colony system for autonomous intersection management. *Applied Intelligence*, *37*(2), 207–222.

https://doi.org/10.1007/s10489-011-0322-z

Yan, F., Dridi, M., & El Moudni, A. (2008). Control of traffic lights in intersection: A new branch and bound approach. In *5th International Conference Service Systems and Service Management - Exploring Service Dynamics with Science and Innovative Technology, ICSSSM'08* (pp. 1–6). IEEE. https://doi.org/10.1109/ICSSSM.2008.4598496

Zha, L., Zhang, Y., Songchitruksa, P., & Middleton, D. R. (2016). An Integrated Dilemma Zone Protection System Using Connected Vehicle Technology. *IEEE Transactions on Intelligent Transportation Systems*, *17*(6), 1714–1723. https://doi.org/10.1109/TITS.2015.2490222

Zohdy, I. H., & Rakha, H. (2012). Game theory algorithm for intersection-based cooperative adaptive cruise control (CACC) systems. In *2012 15th International IEEE Conference on Intelligent Transportation Systems* (pp. 1097–1102). IEEE. https://doi.org/10.1109/ITSC.2012.6338644

APPENDIX A – Literature Review

This appendix provides an overview of recent studies on automated and connected vehicle operations. Although these studies make different assumptions regarding the vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communication, the majority of the articles conclude that autonomous vehicles will have an overall positive effect on intersection operations.

Table A-1 summarizes the articles reviewed along with key elements and approaches used in each research effort. All articles are based on simulation analysis. For simplification purposes, each research effort made a specific set of assumptions:

- The vast majority of the articles did not consider pedestrian presence at the intersection.
- The vast majority of algorithms reviewed assumed a typical four-approach intersection configuration, did not allow lane changing within the communication range, assumed level terrain, and no turning movements.
- From the communication point of view, the process of sending and receiving data among vehicles and infrastructures were assumed to be with no interruptions and no delays.
- Only a few papers considered a mix of automated and conventional vehicles rather than a fully automated vehicle environment.
- Very few studies focused on prevent queue spillback. One study proposed switching priority to the longer queued approach regardless of possible negative effects on total intersection delay.

Table A-1 Overview of Papers Evaluating Traffic Operational Effects of Automated/Connected Vehicle Technology at Intersections

Research	Vehicle Type*	Control Logic**	Spillback Prevention	Lane Changing	Pedestrian	Objective	Simple Geometric Design with Level Terrain	Turning Movements	Synchronized Adjacent Intersections	Perfect Communication Assumed	Additional Consideration	Simulation Platform	Demand Sensitivity Analysis of Proposed Algorithm	Real World Test
(Yan et al. 2008)	A	S				Minimize The Evacuating Time	✓			✓		C Coded Script		
(Agbolosu-Amison, Yun et al. 2012)	A	S				Minimizing Delay	✓	✓		✓		VISSIM Parallel to a C# Program		
(Goodall et al. 2013)	A&C	S	✓			Minimizing Combination of Delay, Number of Stops or Decelerations	✓	✓		✓		VISSIM (A Parallel C# Program Extract Individual Vehicles Information as Simulation Continue)	Better Performance at Low and Midlevel Demand	
(Lee et al. 2013)	A&C	S				Maximizing Throughputs and Total Travel Time, Minimizing Emission, Fuel Consumption	✓	✓		✓		VISSIM & MATLAB	Better Performance at Higher Market Penetration and Volumes	
(Li et al. 2013)	A	S				Minimizing Delay	✓	✓		✓		VISSIM (Emphasized)		
(Li et al. 2014)	A	S				Minimizing Delay Deciding On Vehicles' Trajectory	✓			✓		MATLAB Coded Script CORSIM for Actuated Signal Control	Better Performance at Higher Balanced Demand	
(Zha et al. 2016)	A&C	S				Dilemma Zone Protection Delay Reduction	✓			✓		VISSIM		

Table A-1 (Continued): Overview of Papers Evaluating Traffic Operational Effects of Automated/Connected Vehicle Technology at Intersections

Research	Vehicle Type*	Control Logic**	Spillback Prevention	Lane Changing	Pedestrian	Objective	Simple Geometric Design with Level Terrain	Turning Movements	Synchronized Adjacent Intersections	Perfect Communication Assumed	Additional Consideration	Simulation Platform	Demand Sensitivity Analysis of Proposed Algorithm	Real World Test
(Dresner & Stone, 2008)	A	U				First-Come-First-Served Reservation System	✓	✓		✓			Better Performance at Higher Market Penetration of Autonomous Vehicles	
(Au et al. 2011)	A	U	✓			Prioritizing Older Requests in Queue	✓	✓		✓	Unbalanced Demand			
(Wu et al. 2012)	A	U				Minimizing The Maximum Exit Time of the Last Vehicle Crossing the Intersection	✓	✓		✓	Buses, Vehicles with Less Pollution Can Be Prioritized			
(Zohdy & Rakha, 2012)	A	U				Minimizing delay	✓			✓		MTALAB (Monte Carlo experiment)		
(Ahmane et al. 2013)	A	U					✓			✓				
(Guler et al. 2014)	A&C	U				Minimizing total delay or total number of stops	✓			✓	Unbalanced demand	MATLAB coded script	Better performance at low and balanced demand (because of min green)	
(Tachet et al. 2016)	A	U				Slot-Based System	✓			✓				

* Vehicle Type (A: Automated; C: Conventional human-driven)

** Control Logic (S: Signalized; Unsignalized)

APPENDIX B – Lead Vehicle Trajectory Optimizer Solution

This appendix provides the proposed nonlinear programming formulation for the lead vehicle trajectory optimization (LTO). Even though the problem formulation contains relatively few variables, it belongs to the non-convex optimization class of problems, which are difficult to globally optimize. Due to the need to solve the trajectory optimization for several vehicles, a very efficient solution method is required to conduct the calculations quickly.

Here we note a few observations on the objective function of the LTO problem. The gradient of the objective function can be calculated as:

$$\nabla T(V_2, a_1, a_3) = \begin{pmatrix} \dfrac{a_3(-V_0^2 + V_2^2) + a_1(-2a_3 d_{\text{Comm}} - V_2^2 + V_3^2)}{2a_1 a_3 V_2^2} \\ -\dfrac{(V_0 - V_2)^2}{2a_1^2 V_2} \\ \dfrac{(V_2 - V_3)^2}{2a_3^2 V_2} \end{pmatrix}$$

This shows the travel time is a monotonic function with respect to acceleration or deceleration rates in the first and the last component of each trajectory.

The hessian matrix is the following:

$$\nabla^2 T(V_2, a_1, a_3) = \begin{pmatrix} \dfrac{2d_{\text{Comm}} + \dfrac{V_0^2}{a_1} - \dfrac{V_3^2}{a_3}}{V_2^3} & \dfrac{V_0^2 - V_2^2}{2a_1^2 V_2^2} & \dfrac{V_2^2 - V_3^2}{2a_3^2 V_2^2} \\ \dfrac{V_0^2 - V_2^2}{2a_1^2 V_2^2} & \dfrac{(V_0 - V_2)^2}{a_1^3 V_2} & 0 \\ \dfrac{V_2^2 - V_3^2}{2a_3^2 V_2^2} & 0 & -\dfrac{(V_2 - V_3)^2}{a_3^3 V_2} \end{pmatrix}$$

The hessian matrix is not positive-definite which makes the problem of the non-convex optimization type. One way to simplify the problem is to consider the extreme acceleration/ deceleration rates for the first and the third component of each trajectory. By doing this, two continuous variables, a_1 and a_3, are transformed into substitute variables with values to be either a^{min} or a^{max}.

Therefore, the original problem can be simplified into four sub-problems shown in Table B-1.

Table B-1 Details on the solution of trajectory optimization problem

Optimization	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_4$
Condition	$V_2 \leq V_0 \; and \; V_3 \leq V_2$	$V_0 \leq V_2 \; and \; V_2 \leq V_3$	$V_2 \leq V_0 \; and \; V_2 \leq V_3$	$V_0 \leq V_2 \; and \; V_3 \leq V_2$
Formulation	$\min T(V_2, a^{min}, a^{min})$ $V_2 \leq V^{max}$ Implicit limits on the objective function	$\min T(V_2, a^{max}, a^{max})$ $V_2 \leq V^{max}$ Implicit limits on the objective function	$\min T(V_2, a^{min}, a^{max})$ $V_2 \leq V^{max}$ Implicit limits on the objective function	$\min T(V_2, a^{max}, a^{min})$ $V_2 \leq V^{max}$ Implicit limits on the objective function
Objective function first derivate w.r.t. V_2	$\frac{\partial T(V_2, a^{min}, a^{min})}{\partial V_2} = 0$ $\rightarrow \frac{V_3^2 - 2a^{min} d_0 - V_0^2}{2a^{min} V_2^2} = 0$ $\rightarrow V_2 = \phi$	$\frac{\partial T(V_2, a^{max}, a^{max})}{\partial V_2} = 0$ $\rightarrow \frac{V_3^2 - 2a^{max} d_0 - V_0^2}{2a^{max} V_2^2} = 0$ $\rightarrow V_2 = \phi$	$\frac{\partial T(V_2, a^{min}, a^{max})}{\partial V_2} = 0 \rightarrow V_2^* = \sqrt{\frac{a^{max}(V_0^2 + 2a^{min} d_{Comm}) - a^{min} V_3^2}{a^{max} - a^{min}}}$	$\frac{\partial T(V_2, a^{max}, a^{min})}{\partial V_2} = 0 \rightarrow V_2^* = \sqrt{\frac{a^{max} V_3^2 - a^{min}(V_0^2 + 2a^{max} d_{Comm})}{a^{max} - a^{min}}}$
Objective function second derivate w.r.t. V_2 at the stationary point	Not needed	Not needed	$\frac{\partial^2 T(V_2^*, a^{min}, a^{max})}{\partial V_2^2} = \frac{-1}{a^{min} a^{max}} \sqrt{\frac{(a^{max} - a^{min})^2}{a^{max}(V_0^2 + 2a^{min} d_{Comm}) - a^{min} V_3^2}} > 0$	$\frac{\partial^2 T(V_2^*, a^{max}, a^{min})}{\partial V_2^2} = \frac{-1}{a^{min} a^{max}} \sqrt{\frac{(a^{max} - a^{min})^2}{a^{max} V_3^2 - a^{min}(V_0^2 + 2a^{max} d_{Comm})}} > 0$
If $T^* < G_{p_i}^{min}$	$V_2^* = \frac{a d_{Comm} + (V_0^2 - V_3^2)/2}{a(t_{G_i} + t_{p_i}) + V_0 - V_3} \vert_{a \in \{a^{min}\}/\{a^{max}\}}$		$V_2^* = \frac{a_3 V_0 - a_1 a_3 (t_{G_i} + t_{p_i}) - a_1 V_3 \pm \sqrt{a_1 a_3 \left(a_1 \left((t_{G_i} + t_{p_i})\left(a_3 (t_{G_i} + t_{p_i}) - 2V_3\right) + 2d_{Comm}\right) - 2a_3 \left(d_{Comm} - (t_{G_i} + t_{p_i}) V_0\right) + (V_0 - V_3)^2\right)}}{a_1 - a_3} \vert_{a_1, a_3 \in \{a^{min}, a^{max}\}}$	

From single variable optimization theory, the optimal answer on each case is either one of stationary points making derivative of objective function zero or on the boundary of the feasible interval.