ROADWAY SAFETY INSTITUTE
Human-centered solutions to advanced roadway safety

Implementation of a V2I Highway Safety System and Connected Vehicle Testbed

John Hourdos
Gordon Parikh
Peter Dirks
Derek Lehrke

Department of Civil, Environmental, and Geo- Engineering
University of Minnesota

Final Report

REGION 5

CTS 19-07

University of Minnesota
Driven to Discover℠

The University of Akron

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

SOUTHERN ILLINOIS UNIVERSITY
EDWARDSVILLE

Western Michigan University

Technical Report Documentation Page

1. Report No. CTS 19-07	2.	3. Recipients Accession No.
4. Title and Subtitle Implementation of a V2I Highway Safety System and Connected Vehicle Testbed		5. Report Date April 2019
		6.
7. Author(s) John Hourdos, Gordon Parikh, Peter Dirks, and Derek Lehrke		8. Performing Organization Report No.
9. Performing Organization Name and Address Department of Civil, Environmental, and Geo- Engineering University of Minnesota 500 Pillsbury Dr SE Minneapolis, MN 55455		10. Project/Task/Work Unit No. CTS #2015037
		11. Contract (C) or Grant (G) No. DTRT13-G-UTC35
12. Sponsoring Organization Name and Address Roadway Safety Institute Center for Transportation Studies University of Minnesota 200 Transportation and Safety Building 511 Washington Ave. SE Minneapolis, MN 55455		13. Type of Report and Period Covered Final Report
		14. Sponsoring Agency Code
15. Supplementary Notes http://www.roadwaysafety.umn.edu/publications/		

16. Abstract (Limit: 250 words)

To better prepare for the Connected Vehicle (CV) roadway, RSI has established a CV testbed along a highly crashed section of I-94, building on the Minnesota Traffic Observatory's existing field lab infrastructure. This real-world testbed was designed to implement and evaluate the next generation of vehicle-based freeway safety applications. The priority of this project was to establish the backbone of the sensor communication network and data collection system along the testbed length.

17. Document Analysis/Descriptors Test beds, Connected vehicles, Radar, Cameras, Evaluation and assessment, Data collection		18. Availability Statement No restrictions. Document available from: National Technical Information Services, Alexandria, Virginia 22312	
19. Security Class (this report) Unclassified	20. Security Class (this page) Unclassified	21. No. of Pages 69	22. Price

IMPLEMENTATION OF A V2I HIGHWAY SAFETY SYSTEM AND CONNECTED VEHICLE TESTBED

FINAL REPORT

Prepared by:

John Hourdos
Gordon Parikh
Peter Dirks
Derek Lehrke

Department of Civil, Environmental, and Geo- Engineering
University of Minnesota

APRIL 2019

Published by:

Roadway Safety Institute
Center for Transportation Studies
University of Minnesota
200 Transportation and Safety Building
511 Washington Ave. SE
Minneapolis, MN 55455

ACKNOWLEDGMENTS

The funding for this project was provided by the United States Department of Transportation's Office of the Assistant Secretary for Research and Technology for the Roadway Safety Institute, the University Transportation Center for USDOT Region 5 under the Moving Ahead for Progress in the 21st Century Act (MAP-21) federal transportation bill passed in 2012.

The research team would like to acknowledge RSI for their generous funding of the project. They would also like to thank MnDOT for supporting the research with great interest and additional funding, as well as Brian Kary at MnDOT for his vision and support. The research team would also like to thank the MTO undergraduate students for their effort in reducing video for this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

The Roadway Safety Institute (RSI) supported the creation of a real-world Connected Vehicle (CV) testbed in order to better prepare for future CV applications. The main objectives of the project were:

- **Create a connected vehicle testbed.** Design and deploy a real-world testbed for the implementation and evaluation of the next generation of vehicle-based freeway safety applications.
- **Expand infrastructure-based DMAs to CV-based.** Capitalize on infrastructure-based systems for SPD-HARM and Q-WARN by expanding them to CV-based systems.

The Testbed was established utilizing an existing field laboratory run by the Minnesota Traffic Observatory (MTO). Radar and cameras were deployed to support the existing field lab infrastructure. Further discussion of the existing lab and deployment process can be found in chapters 3 and 4. Data was transmitted by radio back to the MTO where it was reduced and aggregated. Discussion of the MTO's database, as well as software created to read the database, can be found in Chapters 5 and 6.

Establishing the Testbed allowed for additional research to utilize it as well as created opportunities for future research and testing of CV applications. One area of exploration that has already utilized the Testbed is basic safety message (BSM) emulation. Because vehicle to vehicle (V2V) based applications only work when all vehicles transmit BSMs, testing possible solutions cannot be carried out unless all vehicles on the road are transmitting BSMs or some proxy is found. While it is not a perfect substitute for all vehicles broadcasting BSMs, emulating BSMs by gathering the trajectories of all vehicles via radar and transmitting each vehicle's location and speed from the infrastructure side can be utilized for study.

A first test implementation of the CV Testbed instrumentation involved the testing of the Infrastructure based queue warning system with data provided by the CV Testbed radar network. Following, that the basic components of the INFLO applications were tested. Temporal, spatial and heuristic metrics were gathered from the CV Testbed database and paired with a Crash Prone Condition (CPC) algorithm, but alerts to drivers were not implemented as that fell outside the scope of this project. Further discussion of metric gathering and CPC can be found in Chapter 8.

Other applications that were to be studied include Queue Warning (Q-WARN) and Dynamic Speed Harmonization (SPD-HARM). Due to project budget constraints that limited the number of Direct Short Range Communication (DSRC) radios that could be placed along the Testbed, these applications were unable to be tested. In addition, the radars used along the Testbed were to be upgraded, but the manufacturer revealed it did not plan to release them in the US until February 2016, well after the data collection was due to start. The radars used in the Testbed were capable of tracking up to 64 independently moving targets, but congestion along that section of I-94 sometimes exceeded their capacity. Overall, the final product of this project was a fully functional CV testbed uniquely situated to attract freeway safety-oriented V2I and V2V safety application development, implementation, and evaluation projects.

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

Passive interaction between vehicles and fixed transportation system entities such as traffic signals is rapidly giving way to a new paradigm of connected, interacting entities. This enables new forms of data exchange and provides an opportunity to extend the geographic scope, nature, precision, and latency of control within the transportation system. The United States Department of Transportation (USDOT) defines connected vehicles (CVs) as vehicles of any size that are able to “communicate wirelessly with other vehicles and roadway equipment” (Hill & Krueger, 2015). CVs are at least equipped with radios capable of transmitting and receiving information like basic safety messages (BSM), which can be read by infrastructure (V2I communication), other vehicles (V2V communication) or anything that is listening on the proper, secure network (V2X communication). CVs have the potential to change traffic engineering with applications for improving mobility, safety, and environmental impacts.

1.1.1 Connected Vehicles

The purpose of developing connected vehicles is to improve safety and mobility through more effective communication and coordination among vehicles as well as between vehicles and the infrastructure around them. By enabling vehicles to receive information from other vehicles and the infrastructure, drivers should be able to make better-informed decisions about how, where, and when they will drive. This information can take many forms, among which are real-time notices of slowed traffic or dangerous road conditions, more efficient traffic signalization and routing, advised speeds for mobility or environmental reasons, and accident alerts.

Information is transmitted to a vehicle in a couple ways: vehicle to infrastructure (V2I) applications and vehicle to vehicle (V2V) applications. The main difference between V2I and V2V applications is that V2I applications have the main intelligence residing at the infrastructure generating and transmitting targeted messages to instrumented vehicles whereas V2V applications place all intelligence inside the vehicle with the infrastructure only serving as a communication delegation authority and provider of secondary information.

A notable albeit rudimentary V2I application in operation today is the USDOT’s Highway Advisory Radio (HAR). When the road authority wishes to give drivers information pertaining to traffic and road conditions, special signs with flashing lights are activated to instruct drivers to tune into a specific radio frequency on their car stereo. The frequency they tune into is reserved for a non-commercial radio station that, according to the Federal Communications Commission (FCC), may only broadcast “information pertaining to traffic and road conditions, traffic hazard and travel advisories, directions, availability of lodging, rest stops and service stations, and descriptions of local points of interest” (FCC, 2013). Any given station is limited to a maximum signal of 2 mV/m at 1.5 km, thereby allowing information to be specific to a relatively small geographic area.

Currently developed V2V applications include: forward collision warning (FCW), which alerts a driver of a risk of a rear-end collision when cars ahead are stopped or traveling slowly; emergency electronic brake light (EEBL), which notifies a driver of a vehicle ahead braking suddenly; and blind spot warning (BSW)/ lane change warning (LCW), which warns a driver when a vehicle is in the blind spot, making it unsafe to change lanes. The following is a current list of applications by category. More detailed information can be found in the CV website of the U.S. Department of Transportation's (USDOT) Intelligent Transportation System Joint Program Office (ITS JPO) (USDOT, 2018).

- Environment
  1. Eco-approach and departure at signalized intersections
  2. Eco-freight signal priority
  3. Eco-traffic signal priority
  4. Eco-traffic signal timing
- Road weather
  1. Road weather connected vehicle applications
  2. Information and routing support for emergency responders
  3. Enhanced maintenance decision support system (MDSS)
  4. Information for freight carriers
  5. Information for maintenance and fleet management systems
  6. Motorist advisories and warnings
  7. Weather-responsive traffic management
- Safety
  1. Do not pass warning
  2. Emergency electric brake light warning
  3. Intersection movement assist
  4. Lane change warning/blind spot warning
  5. Forward collision warning
  6. Truck forward collision warning
  7. Left turn across path
  8. Vehicle turning right in front of bus
  9. Red light violation warning
  10. Stop sign gap assistance
  11. Work zone warning
  12. Curve speed warning
  13. Pedestrian in signalized crosswalk
  14. Connected vehicle for safety rail
  15. Transit bus stop pedestrian warning

For the purpose of completeness V2X, a third communication channel that is emerging in the CV environment, will be discussed but only briefly. V2X represents communication between a vehicle and some entity that is neither infrastructure nor other vehicle (e.g., OnStar, Mercedes Benz application).

Examples of this include the OnStar driver assistance program, which provides in-vehicle security, navigation, and remote diagnostics and the Mercedes Benz program.

Since connected vehicles depend on the existence of communication through at least one channel — two vehicles communicating with each other or a vehicle communicating with the infrastructure — testing requires this sort of connected environment. A single connected vehicle on the road without any other parties to communicate with is no different than one of today's unconnected vehicles, but with the addition of connectivity, the connected vehicle can be tested and operated.

1.1.2 USDOT Connected and Autonomous Vehicle Program

The USDOT, through the Connected and Automated Vehicle (CAV) program, is assessing applications that aim to realize the full potential of connected vehicles and infrastructure to enhance current operational practices and transform future surface transportation systems management. One foundational element of the USDOT CAV program is the Dynamic Mobility Applications (DMA) program, a set of applications designed to work with connected vehicles to make travel safer and more efficient. The bundles of applications included under the umbrella of DMA (USDOT, 2014a) are:

- Enabling Advanced Traveler Information System (EnableATIS)
- Freight Advanced Traveler Information Systems (FRATIS)
- Integrated Dynamic Transit Operations (IDTO)
- Intelligent Network Flow Optimization (INFLO)
- Multi-Modal Intelligent Traffic Signal Systems (MMITSS)
- Response, Emergency Staging and Communications, Uniform Management, and Evacuation (R.E.S.C.U.M.E.)

In addition, the USDOT has partnered with several cities around the US to deploy a CV fleet and infrastructure and see what real-world results come from CV applications. Ann Arbor, Michigan, began its program (the Ann Arbor Connected Vehicle Test Environment, or AACVTE) in 2012. More recently, Florida, Wyoming and New York deployed pilot sites designed to test CV applications in wider settings.

This project focusses on the DMAs included in the INFLO bundle. INFLO is a bundle of applications that target maximizing roadway throughput, reducing crashes, and reducing fuel consumption through the use of frequently collected and rapidly disseminated data drawn from wirelessly connected vehicles, travelers' communication devices, and infrastructure. This project specifically addresses the Dynamic Speed Harmonization (SPD-HARM) and Queue Warning (Q-WARN) applications.

The two main DMAs focused in this project are Dynamic Speed Harmonization (SPD-HARM) and Queue Warning (Q-WARN) — both of which are in the INFLO bundle.

The INFLO SPD-HARM concept aims to maximize throughput and reduce crashes by utilizing V2I and V2V communication to detect impending congestion that might necessitate speed harmonization, generating an appropriate target speed recommendation for upstream traffic, and communicating the recommendations to the affected. The SPD-HARM concept reflects an operational environment in which speed recommendation decisions are made at a traffic management center (TMC) or a similar infrastructure-based entity and then communicated to the affected traffic. In such an environment, the SPD-HARM application resides within the infrastructure-based entity and is external to the vehicle.

The INFLO Q-WARN application concept aims to minimize or prevent impacts of rear-end or secondary collisions by utilizing V2I and V2V communication to detect existing queues and/or predict impending queues and communicate advisory queue warning messages to drivers in advance of roadway segments with existing or developing vehicle queues. The Q-WARN application may reside in the vehicle, reside within an infrastructure-based entity, or utilize a combination of the two. The queue warning messages may either be broadcast by the infrastructure-based entity using V2I communication or broadcast by vehicles that are in a queued state to nearby vehicles and infrastructure-based entities.

1.1.3 Minnesota Traffic Observatory’s I-94 Field Laboratory

For purposes of research and development related to intelligent transportation systems (ITS), a traffic detection and surveillance laboratory was established in 2002 by the Minnesota Traffic Observatory (MTO) of the University of Minnesota (Hourdakis, Michalopuloulos, & Morris, 2004). The I-94 Field Lab is located on a 1.7-mile-long section of the I-94/I-35W “commons” through downtown Minneapolis. I-94 is a connector-type freeway joining the cities of St. Paul and Minneapolis. At this location, it carries average daily traffic of more than 80,000 vehicles in each direction and is congested for at least 5 hours a day, especially during the afternoon peak period.

Three types of sensors were incorporated in the I-94 Field Lab’s data collection capabilities: cameras and machine vision detectors (MVDs), which are located on the rooftops of three high-rise buildings adjacent to the freeway to collect simultaneous video and traffic measurements, and MnDOT’s inductive loop detectors for collecting 30-second-aggregated volume and occupancy for every lane. To connect all of these stations to the MTO’s local network, a number of radios are used to create wireless bridges between the MTO and all remote hardware.

1.2 OBJECTIVES

The main objectives of the project described in this report were as follows:

- **Create a connected vehicle testbed.** Design and deploy a real-world testbed for the implementation and evaluation of the next generation of vehicle-based freeway safety applications.
- **Expand infrastructure-based DMAs to CV-based.** Capitalize on infrastructure-based systems for SPD-HARM and Q-WARN by expanding them to CV-based systems.

The objectives of creating a testbed for connected vehicles and expanding infrastructure-based systems into the CV realm leveraged the existing I-94 Field Lab's capabilities and expanded on them by adding roadside radar stations along the corridor. The radar stations were spaced along the road so as to provide uninterrupted coverage. The array of radars collected high-resolution vehicle trajectory data for every vehicle that passed through the corridor thereby providing data on the location and speed of individual vehicles for the length of the corridor, not just at the locations with MVDs or loop detectors.

1.3 CURRENT AND FUTURE TESTBED USES

Following the completion of the main objectives of the project, some initial utilizations of the testbed were explored while others were left as available resources for future research and testing.

One intended use for the new I-94 V2I Testbed was the development of an environment for the evaluation of existing and development of new in-vehicle driver warning messages for SPD-HARM and Q-WARN. The system allows for rapid in-vehicle alerts and suggested speeds to oncoming vehicles. Testing of warning messages, for the moment, simply followed the same method used in the USDOT's INFLO demonstration where data exchange between infrastructure and vehicles was emulated by cellular communication. Normally, a proper implementation would require the deployment of several dedicated short-range communication (DSRC) radios along the corridor. Note that DSRC is not the only option for a CAV environment, but at the time of this writing, it was the forefront CAV radio. Other options like Auto5G or Cellular-V2X exist, but both cost and security concerns of letting a third party have access to traffic data make DSRC a more amenable choice of radio. An onboard device (OBD) and a roadside unit (RSU) were acquired and tested, but the project budget did not allow for more DSRC radios to be deployed. Currently, the Testbed provides the tools necessary for supporting human factors studies on V2I and V2V such as the evaluation of the synergy or conflict between in-vehicle and external infrastructure-based messages.

One area of exploration that has already utilized the Testbed is basic safety message (BSM) emulation. Because V2V based implementations of SPD-HARM and Q-WARN only work when all vehicles transmit BSMs, testing possible solutions cannot be carried out unless all vehicles on the road are transmitting BSMs or some proxy is found. While it is not a perfect substitute for all vehicles broadcasting BSMs, emulating BSMs by gathering the trajectories of all vehicles via radar and transmitting each vehicle's location and speed from the infrastructure side has potential. An attempt at a V2I substitute was tested and produced mixed results that will be discussed later in this report.

In an effort to study the underlying principles necessary to develop a V2V queue warning system in an unconnected environment such as the Testbed site, a V2I queue warning system was developed and deployed (Dirks, Liu, & Hourdos, 2016). In lieu of BSMs from every vehicle, the system uses the sensor data to predict crash-prone conditions and warn drivers via message boards mounted on gantries. Once vehicles with V2V connectivity are the norm, the source of the vehicle trajectory data will change from infrastructure-based radars to the BSMs broadcast by vehicles. Similarly, because there are no compatible V2I-capable vehicles on the road at this time, the system does not broadcast any warning messages. In the event that such vehicles emerge, the system will be able to communicate with them.

CHAPTER 2: CONNECTED VEHICLE TESTBED CONCEPT

To better evaluate and develop CVs and CV-related systems, a testbed was proposed. The early concept for the Testbed centered around a facility that leveraged the capabilities, location, and infrastructure of the MTO's existing I-94 Field Lab and could be used to test freeway-related CV applications.

2.1 CONSIDERED CV APPLICATIONS

When considering the specific resources to be provided by the Testbed, the USDOT's list of DMAs was compared to the capabilities and characteristics of the existing I-94 Field Lab. The result was a list of applications that could not be tested, maybe be tested, and definitely be tested.

2.1.1 Non-Supportable Applications

The size of the Field Lab necessitated that all data be collected within the boundaries of the lab. This meant that testing of tools for route planning or logistics that would need to be evaluated on a network level could not be supported. As a result, the following applications were not considered:

- Enabling Advanced Traveler Information Services (EnableATIS)
- Freight Specific Dynamic Travel Planning and Performance (from FRATIS)
- Drayage Optimization (from FRATIS)
- T-DISP (from IDTO)
- T-CONNECT (from IDTO)
- D-RIDE (from IDTO)

Due to the location of the existing I-94 Field Lab, only freeway operations applications could be tested. Applications that specifically pertain to intersections were not considered. Applications in this category are the following:

- Intelligent Traffic Signal System (ISIG) (from MMITSS)
- Transit Signal Priority (TSP) (from MMITSS)
- Mobile Accessible Pedestrian Signal System (PED-SIG) (from MMITSS)
- Emergency Vehicle Preemption (PREEMPT) (from MMITSS)
- Freight Signal Priority (FSP) (from MMITSS)

Applications that gather, manage, and disseminate information during emergencies were also excluded because they were not related to freeway operations. These applications were:

- Incident Scene Pre-Arrival Staging Guidance for Emergency Responders (RESP-STG) (from R.E.S.C.U.M.E.)
- Emergency Communications and Evacuation (EVAC) (from R.E.S.C.U.M.E.)

2.1.2 Potentially Supportable Applications

While three of the remaining applications would require BSM emulation to be evaluated in a non-CV environment, two of them require perfect BSM emulation to function. Those two applications are:

- Incident Scene Work Zone Alerts for Drivers and Workers (INC-ZONE) (from R.E.S.C.U.M.E.)
- Cooperative Adaptive Cruise Control (CACC) (from INFLO)

Because of the known limitations of the radars currently available, perfect BSM emulation was not considered to be viable at this time. The radars that are currently available are able to produce very high-resolution data but they are not able to detect and track every vehicle every time. In the event of the release of radars that can produce data at the same or better resolution but in a more reliable way, BSM emulation would become a more viable option for testing.

2.1.3 Fully Supportable Applications

As already mentioned two applications that are feasible to implement and test at and beyond current levels achieved by USDOT are the QWARN & SPDHARM. Q-WARN aims to provide drivers timely warnings and alerts of impending queue backup. The goal is to reduce shockwaves and prevent collisions as well as other secondary crashes. It accomplishes this by predicting the location, duration and length of queue propagation. This is achieved through harvesting BSMs and other V2V location information transmitted by V2V capable vehicles and utilizes I2V communication for rapid dissemination and sharing of queue information. Dynamic Speed Harmonization (SPD-HARM) aims to dynamically "adjust" and coordinate vehicle speeds in response to congestion, incidents, and road conditions to maximize throughput and reduce crashes. The aim is to reduce speed variability among vehicles to improve traffic flow and minimize or delay flow breakdown formation. Again, it uses harvested BSMs as well as traditional traffic detectors to monitor the traffic stream and utilizes I2V communication to provide recommendations directly to drivers.

SPD-HARM and Q-WARN are technologies that have their roots on already used infrastructure-based equivalents. The two differences are the use of BSM as sources of traffic measurements and the dissemination of the information directly to the vehicle instead over external Dynamic Message Signs (DMS). In fact, both such systems have been implemented on I-94 before one developed by the MTO while both evaluated in earlier projects.

In 2015, FHWA funded Battelle to conduct an INFLO prototype small-scale demonstration in Seattle, WA. The Battelle Team had developed prototype INFLO Q-WARN and SPD-HARM applications and demonstrated the functionality and performance of these applications to U.S. DOT as part of controlled environment tests conducted on roadways around Battelle offices in Columbus, Ohio in May 2014. Battelle demonstrated a system that:

- Collects location and heading data from connected vehicle Basic Safety Messages (BSMs) using both cellular and DSRC communications

- Populates that data in a Cloud database
- Processes the data to determine the beginning and end of congestion1 zones
- Processes the data to determine recommended harmonization speeds
- Delivers Queue Ahead, In-queue and Speed Harmonization messages to drivers on a smart-phone interface.

The SPD-HARM prototype simply translated the existing infrastructure based Speed Harmonization used by the Washington State Department of Transportation (WSDOT) which in extend had emulated the algorithm developed by MnDOT and deployed on I-35W and I-94 in 2009. The Q-WARN prototype was even more simplistic relying on infrastructure sensors or BSMs when available to detect the Back Of Queue (BOQ). In difference to what was initially described, there was no prediction or estimation of the BOQ but it simply updated it on the position of the latest very low speed either from a regular detector or from the BSM of a CV. The MTO deployed in 2016 a much more sophisticated Queue Warning algorithm operating with individual vehicle speeds from infrastructure sensors. Utilizing the same logic but with emulated BSMs from the radar sensors was straight forward.

The reason emulated BSMs from the radar sensors were sufficient for the implementation of Q-WARN and SPD-HARM, in difference to the case of CACC, is that such implementation requires only a fraction of vehicles in the stream to be providing location and speed information so the vehicles “missed” by the radar did not affect feasibility. In general, the requirements for the test implementation of the two INFLO applications are the following:

- Has freeway segment with recurring congestion.
    - The I-94 site not only has recurrent congestion but also has a reliable crash history. Additionally, the front of the queue is a stable geometric bottleneck where almost 100% of the shockwaves start their upstream propagation. This location is the downstream boundary of the CV Testbed allowing the system to be used to track the queue for a full 2/3 of a mile. Admittedly, the FHWA Small-Scale Demonstration in Seattle utilized a 5 mile long segment of the I-5 corridor but given that this project also lacked the instrumented vehicles receiving the I2V messages.
- Existing infrastructure-based queue detection
    - The MTO deployed MN-QWARN at the site in 2016 and it still in operation. More importantly, the aforementioned implementation resulted in the establishment of a two way communication and coordination with the RTMC allowing systems residing at the MTO and in extend the CV Testbed, to access detector information and direct messages to the DMS.
- Existing Variable Speed Limit (VSL) policy and accompanying signage
    - Although MnDOT stopped the VSL system since the advisory speed limits were not followed by the drivers, the signage remained in place till early 2018.
- Installed inventory of Dynamic Message Signs
    - There are two full matrix DMS at the upstream border of the I-94 Field Lab and are used for MN-QWARN and are available for use with the CV Testbed.

- 4G LTE Coverage
    - This was used instead of DSRC communication. The project budget did not allow the purchase and deployment of the large number of radios an only DSRC implementation requires.
- Area subject to varying weather conditions
- Accessible RWIS
    - The MTO maintains a weather station on the 3rd Ave rooftop but the real-time video capabilities allow also manual observation of weather conditions.
- Historical Traffic Data (Baseline)
    - Detailed information regarding the traffic conditions in the CV Testbed started being archived in 2003.
- Emulated BSMs from a sufficient large amount of vehicles in the Testbed.
    - This was the focus of this effort. The design, development, deployment, and testing of the advanced sensor infrastructure that would provide sufficient data for implementing the two INFLO applications

According to FHWA the following are also required for a proper testing of the INFLO applications but such a test was not in the scope of the CV Testbed project.

- Route is a primary route for a targeted participant base.
- Local employer(s) with shift-based schedule
- Local operator familiar with ITS Research

2.2 SYSTEM SPECIFICATIONS

Given the state of the available hardware at the time and the needs of the applications, specifications for the system were developed to provide a target for the selection and deployment of roadside hardware. The feature of the system that required the most engineering to achieve was the ability to collect high-resolution position and speed (trajectory) data from vehicles in the corridor in real-time and with high accuracy. For the purposes of developing a CV Testbed, “high-resolution” was defined as a minimum of 10 measurements per second (10 Hz) to match the frequency provided by DSRC-equipped vehicles broadcasting BSMs. This, along with the requirement for accurate position and speed data, greatly limited the available sensor technologies that could be used to implement such a system. This influenced how many of the other specifications were ultimately met.

In addition to the resolution and accuracy requirements, the testbed also was required to provide continuous sensor coverage over the length of the site. This critical feature was required to achieve either BSM emulation or to implement any of the supportable applications, as large gaps in the coverage would significantly affect the performance and reliability of these applications. Because of the limitations in the available sensors, the requirement of continuous sensor coverage effectively necessitated that a series of multiple sensors would be required to collect trajectory data.

To ensure that the data would be suitable for use in real-time applications and that data from multiple sensors could be reliably merged, the system required a source of high-precision time for synchronizing data between sensors and for applications using the data. Because most of the available sensors do not have a universal sense of time built-in, this required a custom solution.

Finally, to support both historical data collection for analysis and development of safety applications, the system was required to have the ability to share and log all data recorded so that it could be used either in real-time or reconstructed to simulate a real-time system. This meant there would have to be at least one method for inter-process communication (IPC) to allow multiple applications to concurrently access to the data from sensors along with a database for storing historical data in a format that could be used to reconstruct the state of the real-time system. Due to the presence of data originating from multiple sensors, each with a unique orientation and local coordinate system, this system would also have to provide some means for translating the sensor data into a global coordinate system with a common reference frame across all sensors.

The existing I-94 Field Lab featured the video coverage necessary to collect data on crashes and near crashes and to provide visual confirmation of the data received from other instruments on site. MVDs provide spot speeds and headways for each vehicle at several points along the lab site.

To fully meet the functional requirements of the proposed testbed, upgrades to the I-94 Field Lab were necessary. On the hardware side, roadside radar stations were necessary to provide high-resolution trajectory data for vehicles in the crash-prone area (green oval in Figure 2-1). Due to the high cost of the radar units, the minimum number of radars that still provided coverage of the crash-prone area and area immediately upstream were called for. Additional radios to connect the new roadside stations to the rooftop stations were also needed. Because there are no immediate plans to test instrumented vehicles with DSRC capabilities on this testbed, DSRC radios were considered to be an unnecessary expense for the current iteration of the Testbed. Vehicles can still receive information from the infrastructure via cellular connection and, should the need for DSRC radio connectivity arise, stations with the necessary power and data connection would still be available.

Figure 2-1 Proposed CV Testbed location (between 1st Ave S and 20th Ave S) with proposed radar coverage highlighted (green oval).

CHAPTER 3: SITE DESCRIPTION

3.1 EXISTING INFRASTRUCTURE (I-94 FIELD LAB)

3.1.1 Stations

The pre-existing rooftop stations are located on the rooftops of the 3rd Avenue Towers, Augustana Apartments, and Cedar High Apartments – 630 (see Figure 3-5). The rooftop stations not only provide a high vantage point for the cameras and Machine Vision Detectors (MVDs) but also a clear line of sight for radio communications between stations and between the base station on the 3rd Avenue Towers rooftop and the station on top of Moos Tower at the University of Minnesota campus. An example of a rooftop station is pictured in Figure 3-1(a) which shows a radio receiver and two MVDs and in Figure 3-1(b) which shows a radio transmitter and two cameras.

Each station houses a variety of sensors as well as a computer for storing and streaming the sensor data, an Ethernet switch to allow for multiple devices to be connected to the computer and thereby the radio, and power from the building.

(a) (b)

Figure 3-1 Rooftop station on 3rd Avenue Towers

The locations of the stations are shown in Figure 3-5. Note that the 3rd Avenue Towers station is positioned such that cameras can look straight down the road.

3.1.2 Machine Vision Detectors

To get higher resolution speed and headway data than that available from the MnDOT loop detectors in place, MVDs were installed. Two Autoscopes were installed at the 3rd Avenue Towers station and two more were installed at the Augustana Apartments station. The MVDs measure the individual vehicle speeds and headways whereas the MnDOT loop detectors are only capable of providing 30-second aggregated volumes and occupancies with estimated speeds and headways. The MVDs can also be repositioned with relative ease compared to the in-pavement detectors.

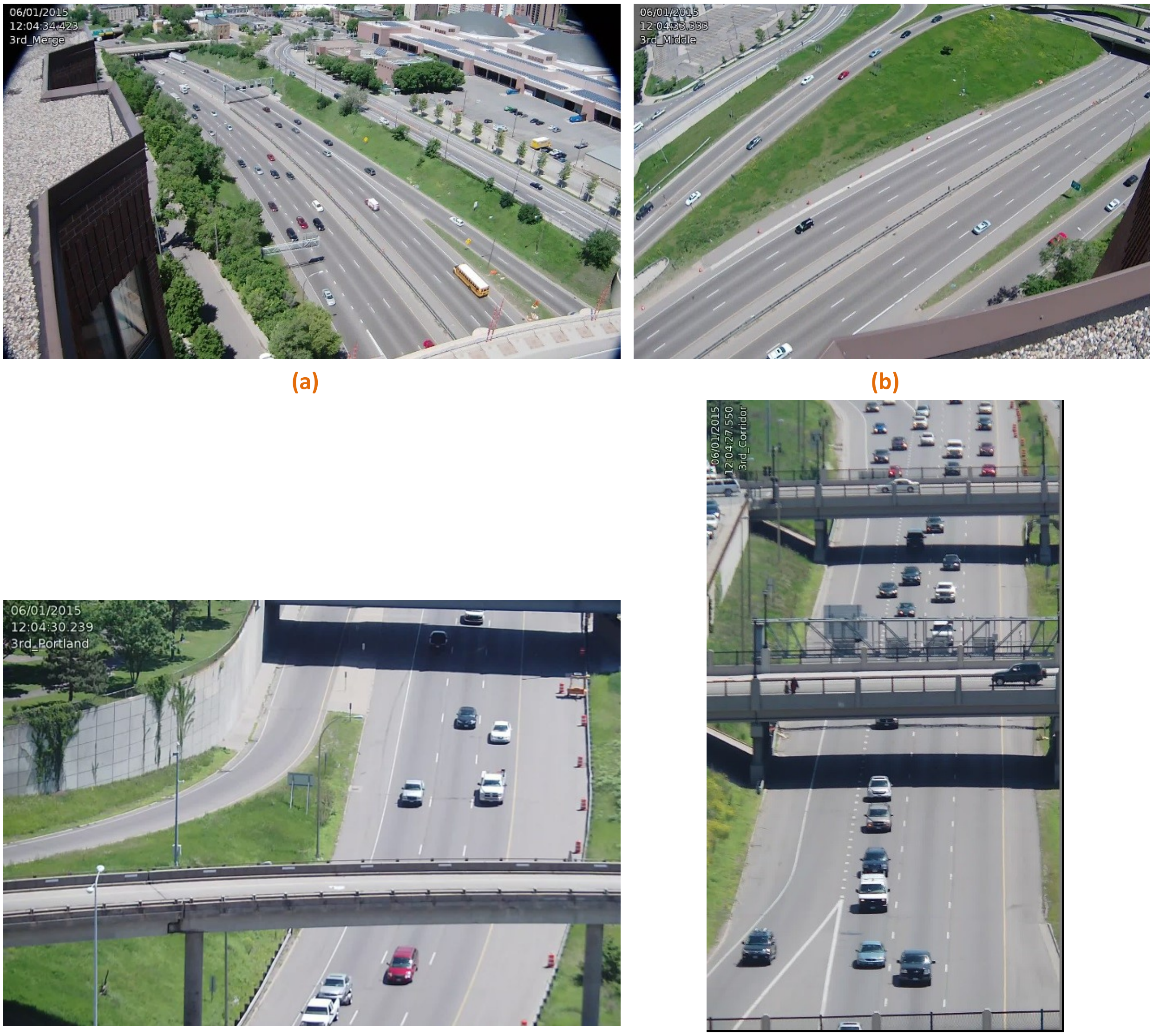
3.1.3 Cameras

As mentioned previously, the rooftop stations provide a good vantage point for recording continuous video of the entire I-94/I-35W commons. Eight IP cameras distributed across the three rooftops provide constant HD video streams of the corridor which are captured by a computer on each rooftop. The area covered by these cameras can be seen in Figure 3-5. Figure 3-2, Figure 3-3, and Figure 3-4 show the field of view of each camera on the rooftops of the 3rd Avenue Towers station, Augustana Apartments station, and Cedar High Apartments – 630 station, respectively, progressing from west to east along the corridor.

3.1.4 Communication Network

Because of the distance between the field sites and the MTO Laboratory in the Civil Engineering Building – ranging from 1 to 2 miles away on the other side of the Mississippi River – the most cost-effective solution for connecting to these sites is radio communication. Advancements in radio technology have allowed such devices to deliver exceptional performance for an affordable price, allowing large wireless networks with enough bandwidth to support streaming multiple HD video feeds and other high-bandwidth activities. The video is saved on location and the recorded file is transmitted later. This increases the reliability of radio communication over transmitting live video, which is possible, but more prone to failure.

As can be seen in Figure 3-6, the majority of the field network hardware provides network access to the three rooftop stations that form the backbone of the I-94 Field Lab. Each of these rooftops contains a number of cameras, MVDs, as well as a recording computer. All network access to the field lab is provided via the primary installation at the 3rd Avenue Towers station. This connection is made with a fiber optic line running from the MTO Laboratory to the roof of Moos Tower, the tallest building on the East Bank campus of the University of Minnesota (Link 1 in Figure 3-6), and a wireless Point-To-Point (PTP) link between Moos Tower and the 3rd Avenue Towers station (Link 2 in Figure 3-6). From there, a PTP link is made to the Augustana Apartments station (Link 3 in Figure 3-6), and another from the Augustana Apartments station to the Cedar High Apartments – 630 station (Link 4 in Figure 3-6). The three rooftop links are made with high-power 5 GHz radios with reliable throughput in excess of 50 Mbps thereby providing a stable and relatively fast connection to the lab at these rooftops. These rooftop stations are permanent.



(a) (b)

(c) (d)

Figure 3-2 Views from cameras at the 3rd Avenue Towers station: Merge (top-left), Middle (top-right), Portland (bottom-left), Corridor (bottom-right)

(a) (b)

Figure 3-3 Views from cameras at the Augustana Apartments station: West (left) and East (right)

(a) (b)

Figure 3-4 Views from cameras at the Cedar High Apartments – 630 station: West (left) and East (right)

Figure 3-5 Rooftop stations (yellow dots) and video coverage by the cameras on the rooftops (outlined in green) of the existing I-94 Field Laboratory

Because the radios all act as bridges, devices on the rooftops appear as if they are in the lab; all that is required to connect a device is plugging it into an available port on the Ethernet switch at its station and obtaining an IP address (either manually or with the main DHCP server). With this network in place, all of the hardware currently in the field can be accessed from the MTO's network without having to know the details of the underlying communication infrastructure. Because each piece in the network operates independently of any other piece, components can be replaced or added without having to change any other part of the network. This note only makes the network easy to use and administer, but it also makes it very extensible. Additional devices can be added to the network by simply plugging the device in to an Ethernet switch and new stations can be given network access by including a radio receiver configured to connect to an available access point.

Figure 3-6 Rooftop stations (yellow dots) connected via point-to-point radio (blue links 3 and 4) to each other and to Moos Tower on the U of M East Bank campus (blue link 2). The Moos Tower station is connected to the MTO via a fiber optic line (blue link 1).

CHAPTER 4: CONNECTED VEHICLE TESTBED INFRASTRUCTURE

Upgrades to the existed infrastructure at the Field Lab consisted of the installation of radar stations at street level and the restructuring of the communication network.

4.1.1 Stations

To support the new needs of the Field Lab, five roadside stations were added (see Figure 4-3). Three types of stations were installed – ITS pole stations, solar pole stations, and gantry stations. The two ITS pole stations – 3rd Ave ITS pole (Figure 4-2(a)) and Portland Ave ITS pole – both existed before the upgrades to the Field Lab and were already wired with power. Similarly, the gantry station – Park Ave Gantry (Figure 4-2(c)) – existed before the upgrades and had power already. The two solar poles – Portland Ave solar pole and Park Ave solar pole (Figure 4-2(b)) – had to be installed and outfitted with solar panels and batteries to compensate for the lack of power access. The locations of the stations were selected primarily due to the presence of existing poles owned by MnDOT. In cases where additional coverage was required, the solar-powered stations were placed in the location that best optimized the sensor coverage based on the locations of adjacent stations, and the safety of the installation in the event of crashes.

The capacity of the solar power stations is adequate for their needs. Of course, the amount of sun they get will affect how high the voltage in the batteries becomes, and the amount of things drawing on the battery affects capacity. Temperature, how long the days and nights are, the angle of the sun depending on the season, cloud cover, and time the solar panels have had to accumulate grime between rainfalls also will affect the units. Except for times when they're covered in snow for extended periods of time or it is too cold for the batteries to work properly, as is the case sometimes in Minnesota winters, the units have not had any issues.

4.1.2 Radar

In contrast to the rooftop sites, which are used for video recording and communication due to their high vantage point, the primary function of the I-94 Field Lab roadside stations is to collect radar data. These stations are much smaller than the rooftop installations, requiring only a few components per station. In addition to the radar sensor, each station includes at least: an AC power adapter for powering the devices, a serial server for making the RS-485 serial data from the sensor(s) accessible over the network, an Ethernet switch, and a radio for communicating back to the access point. In addition to these components, roadside stations can also have a camera as an aid for validating the radar data. The camera is typically a covert camera with a small image sensor (Figure 4-1), allowing the camera to be mounted on the radar.

With the exception of the radar(s), camera (if applicable), and radio, the station components are secured in a weatherized case and (if possible) placed closer to the ground to allow easier access in the event maintenance is required. When the components are functioning as intended, the radio and serial server allow the radar to be accessed as if it was connected directly to the computer to facilitate data collection or development.

Figure 4-1 Radar (large black box) mounted on pole with covert camera (in white housing) and GPS antenna (small black box above radar)

The sensors installed at the stations are 24 GHz radar manufactured by Smart Microwave Sensors (SMS). The sensors are designed for infrastructure applications and have been approved by the FCC for use in infrastructure-based traffic safety applications. While several types of radar were evaluated for this application, ultimately only two types of radar, the UMRR-OA Type 29 and UMRR-OA Type 30 were deployed due to their superior performance in the multilane, high-traffic environment. Specifically, these sensors had the widest field of view of the sensors available at the time, along with reasonable range (distance), as indicated in Table 4.1. While these were the sensors ultimately deployed, there were other sensors manufactured by SMS that were intended to be deployed. Due to ongoing legal issues with importing these devices, however, the research team was unable to procure the sensors within the timeline of the project. The project proceeded with what sensors were available.

Table 4.1 Technical Performance of SMS Type 29 and Type 30 Radar Sensors

Parameter	Type 29 Radar Value	Type 30 Radar Value
Maximum Range on Passenger Car	525 ft	345 ft
Minimum Range	4.9 ft	
Range Accuracy	Larger of < ±2.5% or < ±0.8 ft	
Speed Accuracy	Larger of < ±1% or < ±0.92 ft/s	
Total Field of View (Azimuth)	±18°	±35°

Update Interval	≤ 50 ms
Simultaneously Tracked Objects (Maximum number)	64

The sensor used at each station was selected based on the position of the sensor relative to the roadside and the coverage provided by sensors at adjacent stations. Given the varying field of view and range of the two options, this allowed for nearly continuous coverage in the area with only seven sensors and with a minimum disturbance to the surrounding infrastructure. The sensors deployed at each particular station, along with the type of power used at each installation is indicated in Table 4.2 (Smart Microwave Sensors GmbH, 2017).

Table 4.2 Breakdown of sensors at each station and how the installations are powered.

Station	Sensors	Power
3rd Ave ITS Pole	1x Type 30, Facing Upstream	Infrastructure
Portland Ave Solar Pole	1x Type 30, Facing Upstream	Solar/battery
	1x Type 29, Facing Downstream	
Park Ave ITS Pole	1x Type 30, Facing Downstream	Infrastructure
Park Ave Solar Pole	1x Type 29, Facing Downstream	Solar/battery
Park Ave Gantry	1x Type 29, Facing Upstream	Infrastructure
	1x Type 30, Facing Downstream	

(a)

(b)

(c)

Figure 4-2 Types of roadside stations: (a) ITS pole station with radar and camera mounted on pole and other hardware mounted to the base of the pole, (b) solar pole station with solar panels, radar, and camera mounted on pole and box containing other hardware and batteries mounted to the base of the pole, and (c) gantry station with two radar and camera pairs and enclosure containing other hardware mounted above the road .

Figure 4-3 Rooftop stations (yellow dots), roadside stations (red dots) and radar coverage (red box). The roadside stations and their radars were installed to supplement the existing I-94 Field Lab equipment.

4.1.3 Communication Network

To facilitate the development and eventual deployment of applications that used radar data from the roadside stations, the existing communication network linking the rooftop field lab stations to the lab at the University was upgraded and expanded to provide access to roadside stations. To handle the additional bandwidth, the rooftop stations were upgraded with new radios, with network access to the University provided by a main backhaul between the 3rd Ave Tower in the I-94 Commons area to Moos Tower on the University's East Bank campus. The remaining rooftops connect via dedicated point-to-point (PTP) links that provide optimal line-of-sight between the devices. To connect the roadside stations, a point-to-multipoint radio with a large sector antenna sits on the 3rd Ave Tower with a clear view of the entire corridor, to which radios at each of the roadside stations connect. A diagram of this network can be seen in Figure 4-4.

While this wireless network provides stable network access to the stations that is sufficient for basic administration, development, and testing of the data management system, the use of radio communication leads to high and variable latency that, without correction makes this data unsuitable for time-critical safety applications. Future CV deployments by the MTO or test sites using DSRC should not have this high latency issue. This required the development of a custom solution to account for this variability in transmission time that would ensure that each packet of data received from the radar was given an accurate timestamp that was synchronized with all other sensors in the installation. Because the radar have no sense of time outside of their clock, which resets at each boot, this meant that additional hardware would be required at the stations. Ultimately, each station was outfitted with a GPS-enabled computer with a pulse-per-second (PPS) line that allowed the computer to act as a stratum-0 NTP server, meaning that its clock is accurate to sub-millisecond levels, which is well within the tolerance given the sensor frequency. These computers connect to the serial-to-Ethernet servers directly over the wired local area network inside the stations and timestamp the data packets as soon as they are decoded. Timestamping the packets upon receipt ensures that a coordinated time is present in less than a millisecond which allows the data to be sent over a wireless connection without being affected by the variable latency.

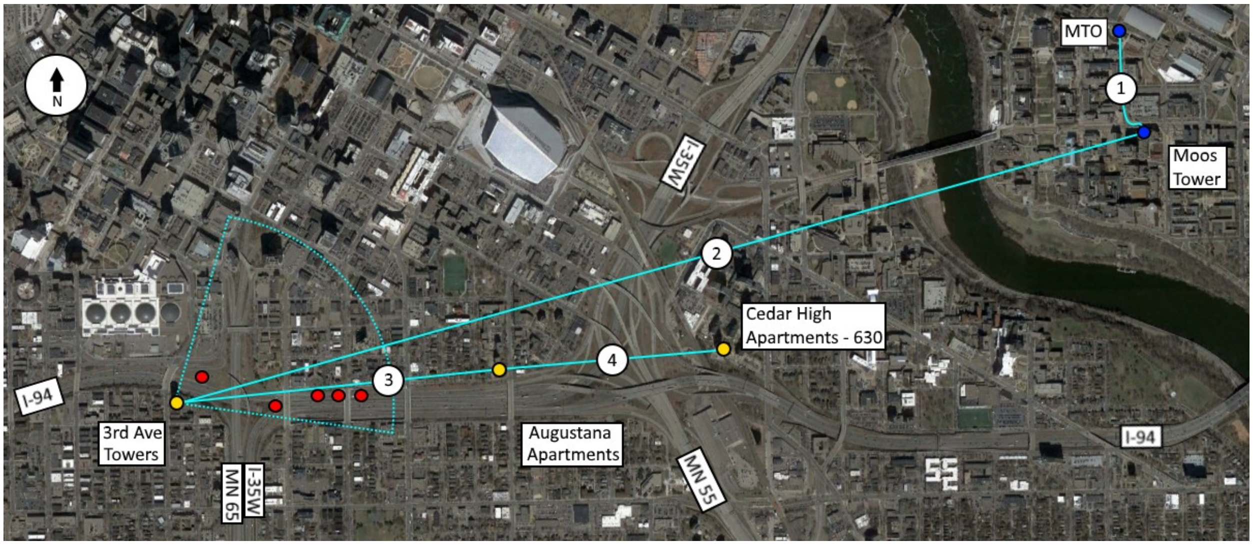


Figure 4-4 Rooftop stations (yellow dots) connected via point-to-point radio (blue links 3 and 4) to each other and to Moos Tower on the U of M East Bank campus (blue link 2). The Moos Tower station is connected to the MTO via a fiber optic line (blue link 1). Roadside stations (red dots) are connected to the 3rd Ave Towers Station – and thereby the MTO – via point-to-many-point radio (dashed blue region).

CHAPTER 5: DATA MANAGEMENT SYSTEM ARCHITECTURE

5.1 TEST OF CV TESTBED HARDWARE

At the beginning of the project, a number of sensor types and manufacturers were considered for collecting the high-resolution data needed to develop the testbed. These included options such as vision, LIDAR, and radar sensors from several manufacturers. Ultimately, it was decided that vision sensors had too many drawbacks to be used for this purpose, including being processing-intensive, being very sensitive to light and weather conditions, and providing less-accurate speed measurements compared to the alternatives. LIDAR was also ruled out, as it could not provide a large enough field of view compared to radar and was considerably more expensive. Previous research had also evaluated the possibility of using 76-77 GHz automotive radar manufactured by Delphi Automotive (now Aptiv), however, as per FCC regulations, these sensors are only allowed for fixed infrastructure use in airport air operations areas, due to the potential to cause interference with automotive radar on vehicles (Federal Communications Commission, 2015). In the end, 24 GHz radars manufactured by Smart Microwave Sensors were selected due to the aforementioned reasons, as well as their wide-scale deployment in infrastructure-based traffic control settings and the additional software resources provided.

During the course of the project, the sensors were tested both in the lab and in the field in a variety of conditions to help researchers design sensor stations for deployment and understand their capabilities and limitations in different types of infrastructure. Initial testing was performed in the lab to develop preliminary software for decoding the data and to assemble stations that would be suitable for temporary deployments. After prototype stations were developed and software was available for preliminary data collection, test deployments were performed as part of another research project that required high resolution trajectory data. The experiences during these temporary deployments were then used to refine the design of the hardware and software that interacts with the radar in anticipation of the final deployment on I-94.

5.1.1 Lab Test

Initial testing of the sensors was performed in the lab to allow development of the custom wire harness that would be used to power the sensors and read the data they produce. This testing also allowed the preliminary development of the custom driver software that would ultimately be used to interact with the sensors in the final deployment on I-94. This initial testing was facilitated by using the sensors' embedded "simulation mode," which simulates a number of vehicles passing through the sensors' field of view with a known trajectory, as well as by using a Doppler simulator provided by SMS, which generates a Doppler-shift signal that is interpreted by the radar similarly to a moving target.

5.1.2 Temporary Deployment/Field Test

After the initial lab testing was conducted, a final design was selected for the temporary field deployments that made up the second phase of testing. For these deployments, the sensors were powered from weatherized plastic boxes containing high-capacity car batteries and mounted on folding sign stands, typically used for holding work zone signs. Small cameras were also included at each station to help validate the data during analysis and were affixed to the stands using high-power magnets. The sensors were wired directly to the stations with data links to the individual stations provided by using radios to connect back to a central command post. At this point in the development, the driver application needed to be manually started from a laptop at the central command post using ad-hoc scripts that were specific to the layout used for this project.

The temporary deployments that constituted the first field tests of a multi-sensor data collection system were performed as part of a separate research project that involved evaluating work zone traffic control layouts on rural, two-lane roads. This involved collecting detailed vehicle trajectories from a distance of over 1,500 feet, making it a good application for testing the performance of the sensors. The weatherized plastic boxes were convenient for the quick deployments necessitated by the rapid pace of the work zone being monitored, where sensors would be deployed for no more than a day before having to move to follow the workers, however there were drawbacks to the system that spurred improvements of the later designs that were developed.

A main drawback to the system as it existed during these first field tests was caused to the fact that sensors were hardwired into the stations using screw terminal blocks, meaning that each sensor was effectively tied to a particular station for the length of the multi-day study. While this eliminated the need to perform delicate wiring in the field, it also required careful treatment when deploying and storing the sensors to avoid tangling the cable or accidentally disconnecting wires. The wiring also varied slightly from station to station, meaning that when issues did arise they were somewhat difficult to debug and fix. In addition to this, the driver application that was used to collect data from the sensors lacked some features that would later prove to be essential to the system, namely the ability to have data collection start automatically. There were also bugs in the software that occasionally caused data to be corrupted or the software to stop working, requiring the software to be restarted frequently throughout the day.

Despite these issues, these field tests were still a good opportunity to learn more about the sensors and test their performance in a real-world setting. This included understanding the true range of the sensors, their sensitivity to orientation, target size, and target speed, as well as the relative performance of the different models of sensors available. They also provided a glimpse into the latency issues caused by the wireless communication that drove the decision to include a synchronized time source, eventually provided by GPS clocks, into the final station design.

After these initial field tests, aspects of the system were altered or redesigned to make the system more modular and easier to deploy. This included: upgrading the power distribution wiring in the stations to make it easier to trace connections and debug wiring issues, developing a quick-disconnect plug for the radars' power/data connector using RJ-45 connectors, and mounting the cameras directly to the radar mounting bracket to give them the same perspective. The driver software was also improved to fix the bugs that caused data corruption or crashes, making the software much more reliable.

5.1.3 On-Site Tests

Further field testing was conducted at locations in the I-94 field lab with easy physical access to the sensor and existing power infrastructure. This allowed researchers to understand the limitations of the sensors in the high-traffic environment in which they would eventually be deployed, helping influence the ultimate selection of the sensors that were deployed as well as the ideal location, spacing, orientation, and number of sensors that would be required. It also gave a sense of what kind of data the sensors would produce in this environment which helped inform which applications could be implemented and the development of filtering algorithms that would allow these applications to function adequately.

During this time, a few more modifications were made to the hardware and software. Weatherproof RJ-45 couplers were used to terminate the connection from the radar, allowing the radar to connect to the station with a standard Ethernet cable thereby making it easier to service or replace the sensor. The GPS clock solution that was devised to deal with the variable latency issue was also developed and added at this point. This phase also saw the majority of the development of the system for collecting data from multiple sensors and making it available for real-time applications running in the lab. The final system as it currently operates is described in detail in the next section.

5.2 OVERVIEW OF FINAL SYSTEM ARCHITECTURE

The system as it currently exists relies on several software components developed on top of the stations running in the field designed to operate over the wireless communication network. This software serves to abstract these features of the system from the perspective of applications so that they have access to sensor data without needing to worry about details like the serial communication protocol used by the radar, time synchronization across the wireless network, or addresses of hardware in the field. Applications can access the data from all sensors in the system through one of two protocol endpoints running on servers in the lab which are chosen depending on how tolerant the application is to delay in the data received. Regardless of the protocol that is chosen, applications receive the same data.

Data from each sensor is initially captured by computers connected to the wired local area network inside the station. From there, it is decoded from the device's native binary communication protocol by a driver application developed by project researchers. At this point, the data is given a timestamp immediately after it is decoded. The time used for the timestamp is set using an extremely accurate GPS clock. This ensures that the timestamp is synchronized across the entire system, allowing data from different sensors to be mixed while maintaining a universally-true time reference. Once the data is decoded and timestamped, it is broadcast to a shared memory channel that is accessible by any application running on the local computer. This memory channel is then synchronized with a central server in the lab that can be accessed by applications that need access to the data as soon as it is available. In addition to this, another application on the station computer transmits the data over a TCP/IP socket to a server in the lab which then writes the data to a PostgreSQL database. That database makes the data accessible by other applications in a simpler format than the shared memory channels at the cost of a variable delay on the order of a few seconds. A diagram of this system can be seen in Figure 5-1.

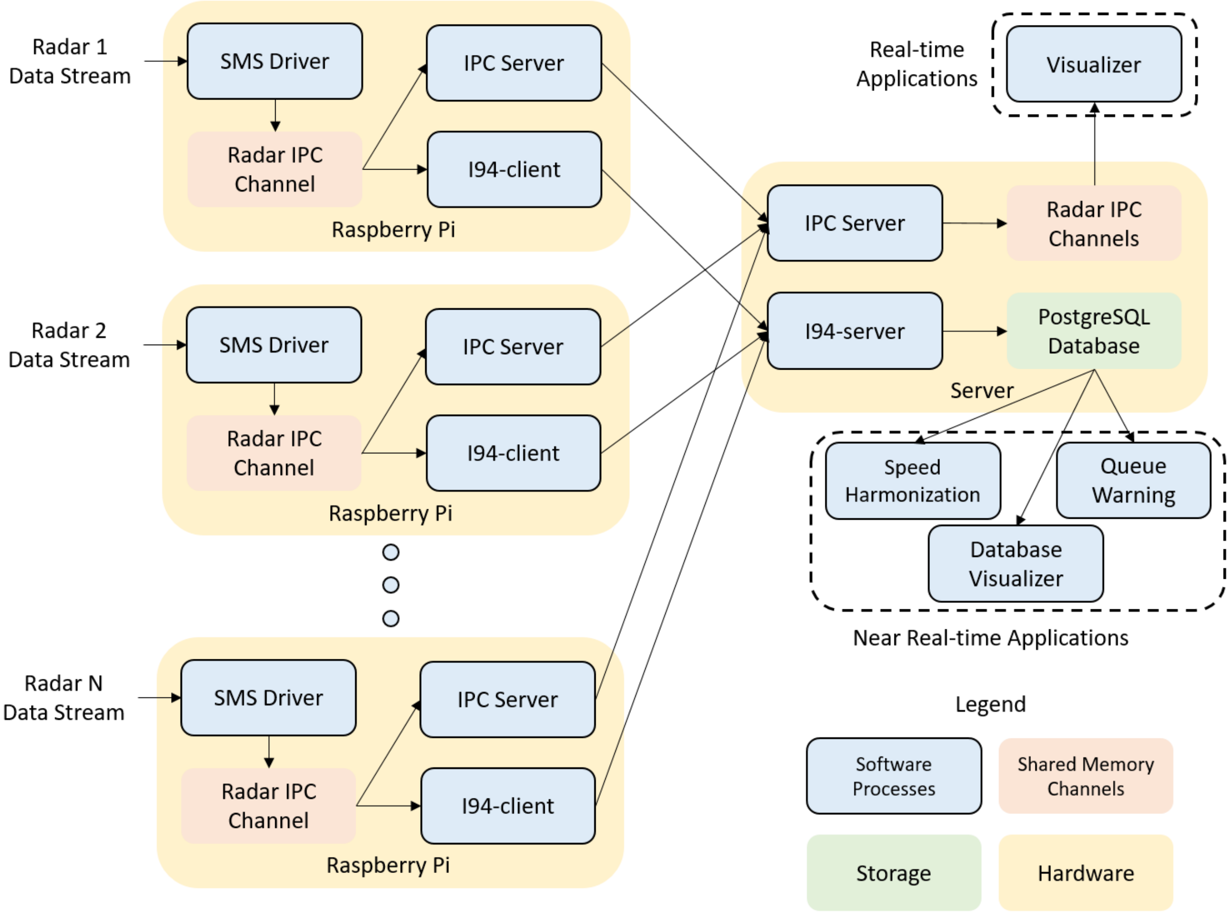


Figure 5-1 Map of system

5.3 SENSOR DRIVER APPLICATION

The driver receives the raw data stream from a sensor and parses or decodes the stream from the binary SMS communication protocol to a format that is more easily understood. The driver is capable of establishing a connection with a sensor over an RS-485 serial connection, or over a TCP connection. The sensor natively outputs the data over serial, but a commercial, off-the-shelf serial-to-TCP adapter is used to send the data over a network. This is advantageous both because it allows a low-level connection to the sensor over a network, which is required to configure the sensor, and because it eliminates the need for additional specialized hardware in the stations that would otherwise be required to connect the sensor to the computer.

The main loop of the driver continually listens for data and then reads it a single byte at a time into a buffer. This read call will block for up to 5 seconds to receive this data, and otherwise exit if nothing is received, assuming there is an issue with the radar or the connection. The buffer is parsed to look for the markers that that signify the beginning or end of a single frame, the data constituting a single sensor measurement. Once the parser has identified it is within a message, it looks for sub-messages that contain the actual data, identified by the corresponding CAN ID (referred to as ident in the documentation). Once a complete sub message is identified, it is sent to another function that reads the ident to determine the type of data it contains, then decodes that data. This is continued until the stop sequence is received signifying the end of the message frame, after which the entire process repeats.

The sensor sends information about its own status as well as position information about the targets in its view. After being parsed, this data is stored in a global variable that is populated as the message is decoded. Once the message is fully decoded (i.e. the stop sequence is identified), a function is called that prints the data to screen and/or writes it to an inter-process communication (IPC) channel. This IPC channel is a shared memory space that is accessible by other processes running on the local computer, allowing any number of independent applications to use the data as it is read and decoded. Each sensor requires its own driver instance which then writes to a dedicated IPC channel. Multiple driver instances can be run on the same machine and applications can read from any number of IPC channels at a time. This modularity makes it easier to manage a variable number of sensors at each station and in the system as a whole, allowing hardware and software to be added or removed as needed for expansion and maintenance.

5.4 INTER-PROCESS COMMUNICATION (IPC) FOR EXCHANGING DATA

The IPC system used by the driver and other software components uses a proprietary library developed at the University of Minnesota, specifically for decoding and sharing sensor data for time-critical safety applications. This library is general with regard to the sensor used and the data it produces, with the data produced by each application specified in an XML-syntax configuration file. The data fields used for radar data, along with their data types, are listed in Table 5.1.

Table 5.1 Sensor data fields available in IPC data

Field Name	Data Type	Description
time	Double-precision float	Unix timestamp with microsecond resolution
count	Unsigned integer	Cycle count for the radar
status	Integer	Status indicator

targets	Integer	Number of targets in message (up to 64)
id[64]	(Array) Unsigned short integer	Array to hold ID of each target
x[64]	(Array) Float	Array to hold X position of each target
y[64]	(Array) Float	Array to hold Y position of each target
vx[64]	(Array) Float	Array to hold X component of velocity of each target
vy[64]	(Array) Float	Array to hold Y component of velocity of each target
relays[16]	(Array) Boolean	Array to hold status of relay triggers

By abstracting sensor data in this way, the IPC system facilitates quicker development of applications using the data, allows multiple applications to use that data without the potential to interfere with one another, and provides a number of tools to extend the functionality of the system without requiring additional development. While the driver application itself only runs in real-time, making applications responsible for buffering data themselves, the IPC framework comes with utilities for saving data into a binary file and replaying it later. This feature is useful for development, testing, and debugging in that it allows applications to use sensor data without requiring an actual sensor thereby allowing specific conditions to be recreated without complex testing setups.

In addition to this capability, the IPC framework also provides a server application to bridge shared memory channels over a network. It does this by reading messages from a local IPC shared memory space and then sending that information over the network to another instance of the IPC Server which then writes those same messages to that machine's local IPC shared memory. This makes the messages available on that machine as if they had been generated locally, allowing data from multiple sensors to be combined for a single application. Compared to accessing data through the database (described in a later section), this method prioritizes rapid transmission of the latest data for time-critical applications that would rather have the latest data available as soon as possible, discarding intermediate data that might arrive late due to network disruptions.

5.5 CLIENT-SERVER APPLICATIONS FOR LOSSLESS TRANSMISSION OF DATA

The alternate method for accessing data from the sensor network uses additional software components to provide better fault tolerance in the event of network disruptions, making all data available to the application in the order it was sent. To achieve this, this system adds an additional client application on the station computer that reads messages from the IPC channel and transmits it to a server application running on a computer in the lab. These two applications communicate via a TCP socket which uses a timeout to make the applications aware in the event the connection is disrupted. In the event the connection is lost, the client enters a buffering mode which writes data to a file on the disk that can then be transmitted back to the server once the connection is restored, ensuring that no data is lost as long as the stations are operational.

The client software consists of two main threads linked with a thread-safe queue. The first thread continually reads IPC to get each message created by the driver. Similar to the driver, it performs a blocking read for 5 seconds after which the program exits, assuming a radar failure. This message is then repackaged and passed to the other thread via the queue, which then transmits it to the server by writing to the socket. Messages have the following structure, which contains multiple delimiters to simplify the parsing of messages:

([time],[sensor_id],[count],[targets]$[id1],[x1],[y1],[vx1],[vy1],[length1]…

$[idn],[xn],[yn],[vxn],[vyn],[lengthn])

Note that the line break is only included here to increase readability. The messages generated by the client have no line breaks or spaces.

After sending the message, the server is expected to respond with the count value of the message passed to ensure that the message was received. In the event the response is incorrect or not received, or if the write operation fails, the client enters an offline mode, where data is written to a file on the disk so it can eventually be saved.

The server application is implemented using a threaded TCP server that creates a new handler thread for each connection established by a client. Data that is received from the clients is then passed to a database writer thread via a queue. For each message received by the database writer, a query is constructed to insert the data into a PostgreSQL database, from which the data can be accessed by other machines on the network. This allows applications to access sensor data without needing the IPC library or message specification. While this process happens in real-time, there is a variable delay due to the additional processing steps that means that the most recent data available at any given time is on the order of a few seconds old. Applications that can tolerate this small delay, however, generally require less development and are able to be written using an interface that allows them to be run using near-real-time or historical data without modification.

CHAPTER 6: APPLICATION-PROGRAMMING INTERFACE

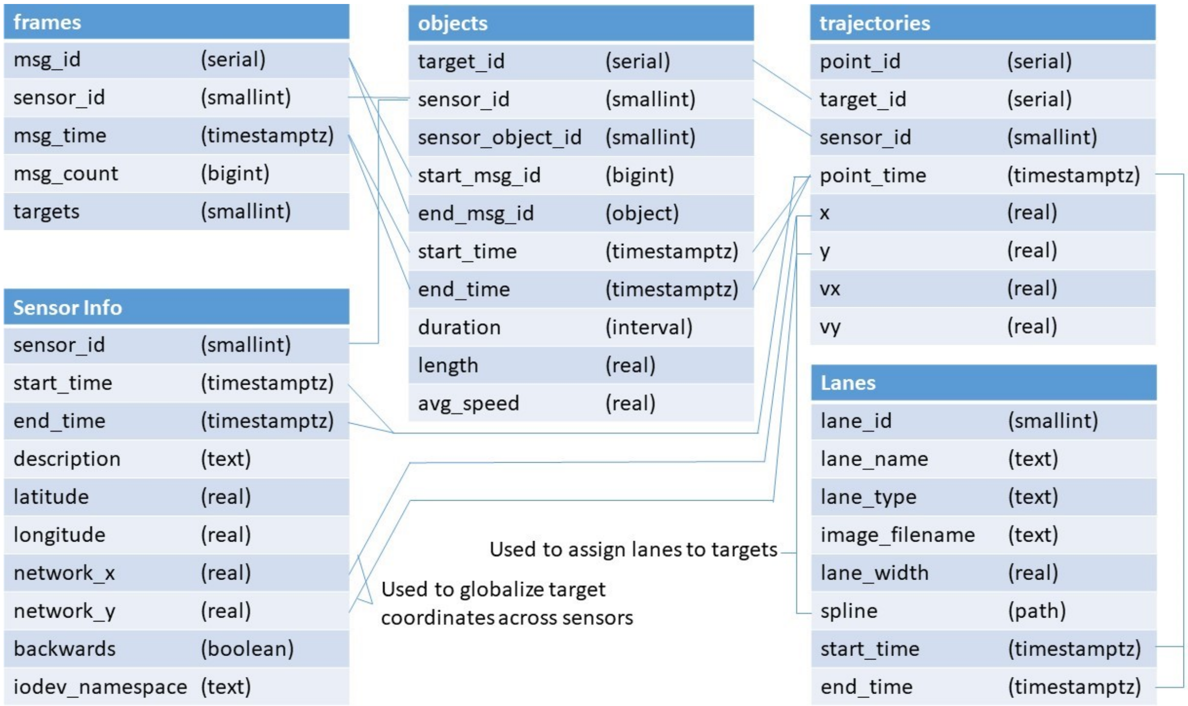
6.1 DATABASE FOR HISTORICAL AND NEAR-REAL TIME APPLICATIONS

The primary interface for applications to access sensor data is via the PostgreSQL database. This database provides access to near-real-time data, generally around one second old, and all of the historical data that has been collected from the sensors. This allows applications to be developed without needing a specialized library or having knowledge of a specific message format. In addition to this, applications can also be developed to work with near-real-time data or historical data without any modifications, allowing applications to be tested easily and using the same code that would run in real-time. Applications can access this data with any PostgreSQL client using SQL statements to read the data into their applications, or by using a set of tools developed in Python designed to read this data.

6.1.1 Database Structure

Data from the sensors is stored in a single PostgreSQL database in real-time using the system described in the previous chapter. The data is saved exactly as it is output by the sensor driver without modification to preserve the original state, allowing further processing to be developed and tested without changing the raw data. In addition to this raw data, the database also contains historical and current position and orientation data for each of the sensors, providing an automatic means for combining data from multiple sensors into a single reference frame. Finally, there is a table that contains lane definitions for the corridor in the reference frame of the sensors, allowing targets to be placed into a lane for analysis purposes.

Because single sensor will produce several million target measurements in a single day, data from the sensors is split into individual tables for each date to reduce the maximum query time. Data is organized into three tables for each date to provide different levels of resolution to analyze the data. At the highest resolution available, in the “trajectories_<date>” table, each record consists of a single target measurement, with an X and Y position and velocity value for each target at each point in time. One level further out, in the “objects_<date>” table, each record consists of a single target as viewed by one sensor over its entire life, containing information like the time of its first and last measurements, the estimated length of the target, and its average speed. Finally, in the “frames_<date>” table, information about the sensor frames that were decoded is provided, including the message count from the sensor and the number of targets observed during that instant in time (up to 64). Together, these tables provide a complete picture of the data read from the sensor while reducing the amount of redundant information stored by the database. A diagram showing these tables and their relationships with each other is shown in Figure 6-1.



frames	
msg_id	(serial)
sensor_id	(smallint)
msg_time	(timestamptz)
msg_count	(bigint)
targets	(smallint)

Sensor Info	
sensor_id	(smallint)
start_time	(timestamptz)
end_time	(timestamptz)
description	(text)
latitude	(real)
longitude	(real)
network_x	(real)
network_y	(real)
backwards	(boolean)
iodev_namespace	(text)

objects	
target_id	(serial)
sensor_id	(smallint)
sensor_object_id	(smallint)
start_msg_id	(bigint)
end_msg_id	(object)
start_time	(timestamptz)
end_time	(timestamptz)
duration	(interval)
length	(real)
avg_speed	(real)

trajectories	
point_id	(serial)
target_id	(serial)
sensor_id	(smallint)
point_time	(timestamptz)
x	(real)
y	(real)
vx	(real)
vy	(real)

Lanes	
lane_id	(smallint)
lane_name	(text)
lane_type	(text)
image_filename	(text)
lane_width	(real)
spline	(path)
start_time	(timestamptz)
end_time	(timestamptz)

Figure 6-1 Database diagram depicting schema of trajectory data from sensors.

In addition to the data tables, there is another table containing both historical and current sensor position and orientation information. In this table, each record contains the sensor ID, the latitude, and longitude of the sensor, the X and Y coordinates of the sensor relative to the first (most-downstream) sensor in the network, and the start and end time for which this configuration is valid. For the current configuration, the end time is left blank to indicate that it is still in place. Additionally, the direction of the sensor (upstream or downstream) is also included, since it is necessary to understand the coordinate system of the sensor relative to the others; while the sensors contain their own orientation configuration that is used to translate ranges and angles into XY coordinates internally, the direction relative to moving traffic is still needed.

The table containing lane information uses an integer ID for each lane, with lane 0 starting at the right-most edge of the road. Lanes are also given a name and designated as either a main or auxiliary lane, as one of the lanes in the corridor exits midway through the sensor installation. The geometry of the lanes is indicated by a spline representing one edge of the lane along with a lane width. This allows targets to be sorted into a lane at each point in time by measuring their distance from the spline and determining if it is within the lane, based on the lane width. Lanes are also defined for a period of time, which allows the system to accommodate changes in road geometry that would occur due to construction.

To simplify the use of the sensor position/orientation table and lane information table for adding information to raw sensor data, a number of SQL functions are also included in the database. These functions use the sensor positions/orientations to transform target coordinates into a global reference frame that is uniform across the network of sensors, and use the lane information to assign a lane to each target at each point in time based on that translated position. This allows users to merge the data from multiple sensors and add this lane information in a single query without having to worry about the details of where the sensors or lanes were located at any particular point in time.

6.1.2 Code for Reading from Database in Real Time

To help software developers and researchers work with the radar data in the database, a set of Python tools have been developed to read this data under a number of circumstances and perform some of the post-processing that would be required by most applications. These tools are optimized to work in real-time (or simulated real-time), reading data from the database in a separate process to make it available to applications without making the main process wait for the relatively slow query operations. They also abstract the process of reading data from the database, instead providing access to the data via a virtual sensor object that translates raw data into objects with an intuitive application-programming interface. This allows application developers to write applications that use sensor data without needing to write queries for reading the data themselves.

A split-process model is used to implement the virtual sensor, where one process reads data from the database in chunks and puts it into a queue that can communicate across processes while another process reads from that queue and constructs objects for the application to access via the virtual sensor. To handle data from multiple sensors, the application can use a sensor network object that further abstracts the network of sensors to behave like a single sensor with a single reference frame. Using data from a configuration file, the sensor network transforms target data as it is read from the database to put it in the reference frame of the downstream-most sensor and assigns lanes to the targets at each point in time. Applications simply call the read method on the sensor network and are provided with a list of targets observed by the sensor network at that instant in time.

By default, this system reads data in real-time as it is inserted into the database however, by providing a start time, it can seamlessly read historical data and play it back as if it is occurring in real-time. This can help applications in the development and testing phases by providing a simple means for testing the application without changing the code that is used. The system can also run faster than real-time, allowing for rapid testing of different applications whose performance can be analyzed after the fact using statistics generated during the run.

As it is currently implemented, the system does not perform any additional filtering on top of the target transformation to perform actions such as following a target as it traverses the individual sensor boundaries and stitching the trajectories together. Once a suitable algorithm is available, however, this functionality could be integrated into the existing framework to similarly abstract that process as well. The split-process model could also be expanded to offload these additional calculations to helper processes thereby allowing the application to utilize all available processing power in the main process for its own calculations.

6.1.3 Applications That Can Use This Data

Based on measurements in the lab, data typically makes it into the database within about one second of it being read from the sensor in the field, though this can vary depending on network load. While in the context of the radar update frequency this is a long time, for the purposes of most applications it is fast enough. Applications that can tolerate this delay therefore can benefit from the simplified API employed by the database and the Python tools provided to accelerate development. This can include applications like QWarn and SPD-Harm, which can function with vehicle-level data at a refresh rate of 5 seconds (Balke, Charara, & Sunkari, 2014).

6.2 IPC FOR TIME-CRITICAL APPLICATIONS

For applications that require the most recent data at all times, an alternate interface for accessing sensor data is provided. This is done via the IPC (inter-process communication) system described in Chapter 5. Though this method requires using a specific library that generally needs more development to integrate into an application, the fact that it works on a lower level and has less intermediate processing means that the data it provides will generally be the latest produced by the sensor. This can be important for time critical safety applications that are required to respond to events within a strict time limit of less than one second.

6.2.1 Applications That Would Require This

In the context of the I-94 installation, the majority of the applications that would require access to the IPC system generally involve some sort of V2V communication emulation. Applications like emulation of Basic Safety Message (BSM) Part I transmission, where messages must be emitted roughly every 100ms, would need to use the IPC system in order to meet this specification requirement. Depending on the computational requirements of the application, such an application could reside on the same computer used to collect radar data in the roadside stations or it could use the IPC server application to transmit data from one or more radar to a computer with more processing power.

6.2.2 Model for Building Applications

As discussed in the previous chapter, applications using the IPC system to access radar data must have the corresponding XML configuration file containing the message specification. In addition to this, applications must also integrate the IPC library into their code, which is required to decode the binary protocol used to encode the data in shared memory. Applications that run on the same computer being used to read data directly from the sensor are able to connect directly to the shared memory channel on that computer however, for applications that require additional computing power or data from multiple sensors, the IPC server application must be used to synchronize these channels over the network.

Under this design, the system will always provide the application with the latest message available from each sensor as fast as it can be written to shared memory (and transmitted across the network, if that is required). This means that the application always has the latest data, however in the event of brief network disruptions that might cause messages to be delayed, old messages could be overwritten by newer messages before they can be read. In a low-latency system, however, this may be desirable, as applications with a strict time limit would generally prefer to act on the most recent data, rather than waste time on information that may be outdated. Because of this, additional constraints are placed on the application that may require some physical aspects of the system, such as the communication network and location of servers for running applications, be redesigned to meet the specification of the application.

6.3 VISUALIZER

For both the IPC and database interfaces, the first applications that were developed to demonstrate the functionality were visualizers. These are simple, graphical programs that display data from multiple sensors in a plan view, allowing users to see the data that is collected from the radar in an intuitive way. The programs show that sensors were either working or not during any selected time periods. The IPC-based viewer is very rudimentary, only combining the data from multiple sensors and compensating for the position and orientation of sensors. The database viewer, by comparison, is more feature-rich with the ability to overlay the data on a map or aerial image, sort objects into lanes based on a predefined configuration, and replay historical data.

6.3.1 IPC for Time-Critical Applications

The visualizer is a program that displays the most recent sensor measurements by reading information over IPC. It is capable of plotting the positions of the tracked targets and could also be used to plot additional information that is generated by the processing algorithm thereby providing a means to verify the operation of sensors and any applications. The IPC visualizer has a simple interface, only showing targets plotted on a white background and colored by sensor (seen in Figure 6-2).

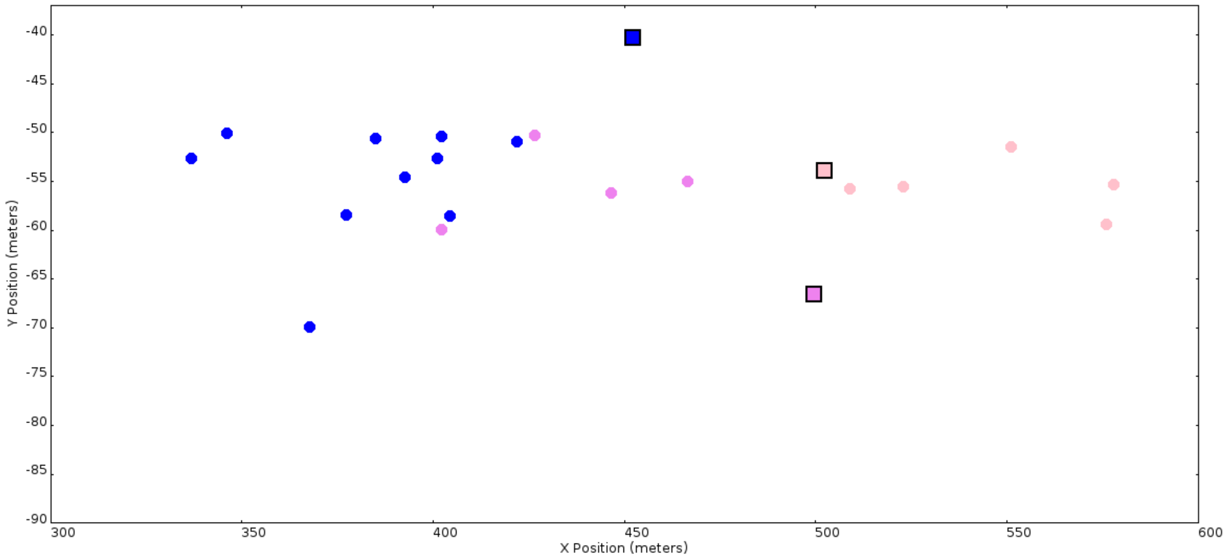


Figure 6-2 Screenshot of the IPC visualizer application

The viewer requires a configuration file to define the locations and orientations of sensors in the field. Data from sensors is read from local IPC channels meaning that, for real-time operation, the IPC server must be used to synchronize data across the network if it is running on a remote computer and to include data from multiple sensors. Because the IPC system also allows data to be recorded and played back later, the visualizer can also read data as it is replayed from these files though it is unable to control the playback.

6.3.2 Database Version as Model for Database Development

Compared to the IPC visualizer, the database visualizer is much more feature-rich. In addition to playing the locations of targets in near-real-time as data is written to the database, it also provides the ability to seamlessly replay historical data by simply providing a start time in the past. Data is also rendered over an aerial image of the site, which helps provide feedback to the user regarding the operation of the sensors (seen in Figure 6-3). It is designed for smooth playback of data, so it will occasionally pause to buffer data but this provides a more visually-appealing experience. In addition to this, the viewer allows the user to pause playback at a particular point in time and to record snapshots or video clips of the viewer output.

Figure 6-3 Screenshot of the database visualizer application, with targets and lanes over an aerial image

This visualizer also requires a configuration file containing sensor positions and orientations but this file contains the same information used to adjust target positions in the database and can also be used to update this information in the database using a simple tool. Like the database, the visualizer also contains definitions of lanes which can be edited by the user using an associated graphical tool, and can sort vehicles into lanes as it plays. Targets can be colored either by the sensor that measured that data or by the lane that is assigned to them.

CHAPTER 7: VALIDATION

7.1 COLLECTION OF DATA FOR VALIDATION

To validate the system, a number of test runs were performed using an instrumented vehicle equipped with a GPS receiver and OBDII vehicle speed logger. The GPS receiver used was a Savari Networks DSRC Onboard Unit, which provides SBAS (satellite-based augmentation system)-corrected position measurements every 100ms. The OBDII logger also provided speed measurements every 100ms. During the runs, radar data was recorded using the data management system discussed in chapter 5, along with video data from all cameras in the corridor. To help identify the vehicle in camera images, the vehicle was marked using paper and painter's tape. Video from the cameras (Figure 7-1) was synchronized with GPS and radar data by measuring the time offset between the video and GPS data when the test vehicle was at a known location along the road.

An important aspect of the system concerned the performance of sensors in collecting vehicle measurements from vehicles driving in any of the lanes in the corridor. To allow this to be evaluated in the analysis, the validation runs consisted of two runs in each of the three lanes in the corridor. Because the corridor can experience significant congestion during both the morning and evening peak periods, the test runs were carried out beginning at 10 AM on a weekday, once congestion had cleared and traffic conditions were normal. Traffic conditions were free-flow during most runs, however given the frequent occurrence of shockwaves in the corridor, which are common in the conditions where the system would be useful, a point was made to collect test data when the vehicle was subject to a shockwave. A notable shockwave was encountered during the last run, providing some data that can be used to evaluate the performance of the sensors in these conditions.

7.2 ANALYSIS OF VALIDATION DATA

The following figures (Figure 7-2 to Figure 7-5) present the result of the CV Testbed Radar system Validation. As can be readily seen on the figures, the Spatial accuracy of the radar system is adequate for the purposes of CV V2I applications but given that a significant number of holes in the detection are observed, it can be surmised that with this kind of radar units, emulating the BSM for V2V applications is not feasible. Additionally, since these radar units are operating based on the Doppler principle, they are unable to detect stationary vehicles. They also tend to confuse targets that are closely spaced in regard to their distance to the radar and with similar speeds, even if they are in different lanes.

Nevertheless, as will be discussed in the following chapter, the amount, resolution, and accuracy of new data this system is providing is still unique.

Figure 7-1 Video views of Test Vehicle on the Leftmost lane during Validation Experiment

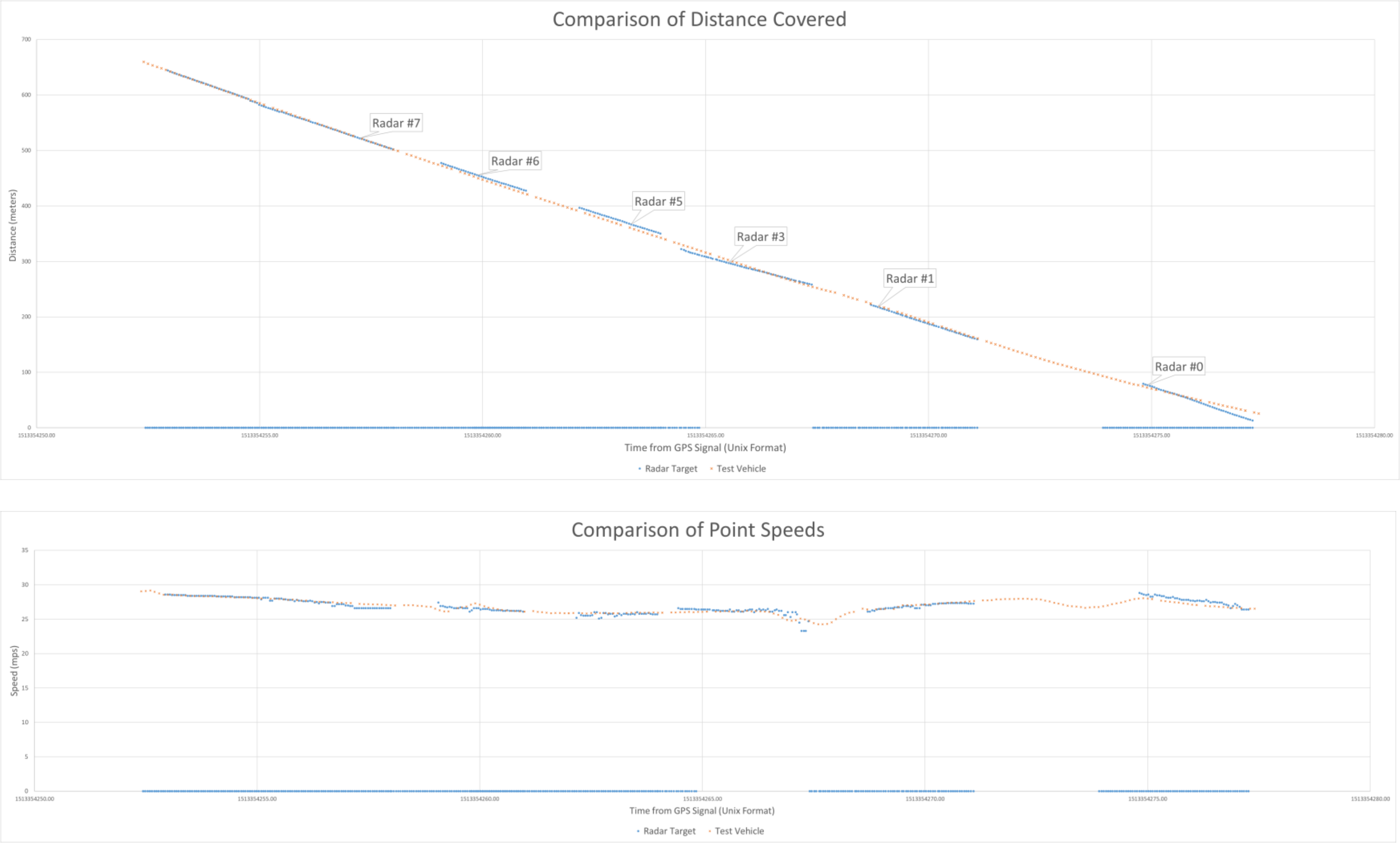


Figure 7-2 Comparison of Radar Measurements with Test Vehicle OBD Position and Speed. Left Lane (First Pass)

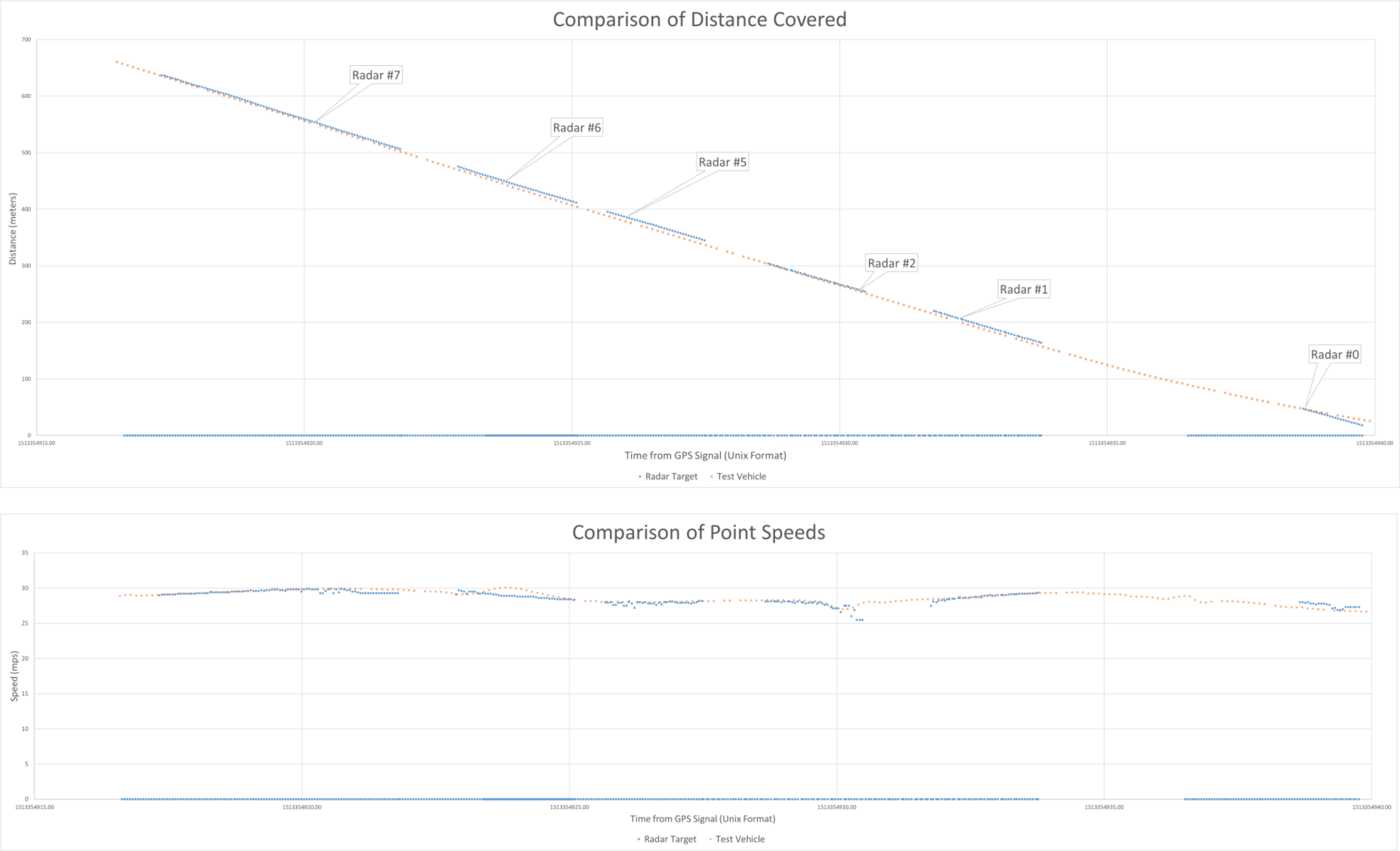


Figure 7-3 Comparison of Radar Measurements with Test Vehicle OBD Position and Speed. Left Lane (Second Pass)

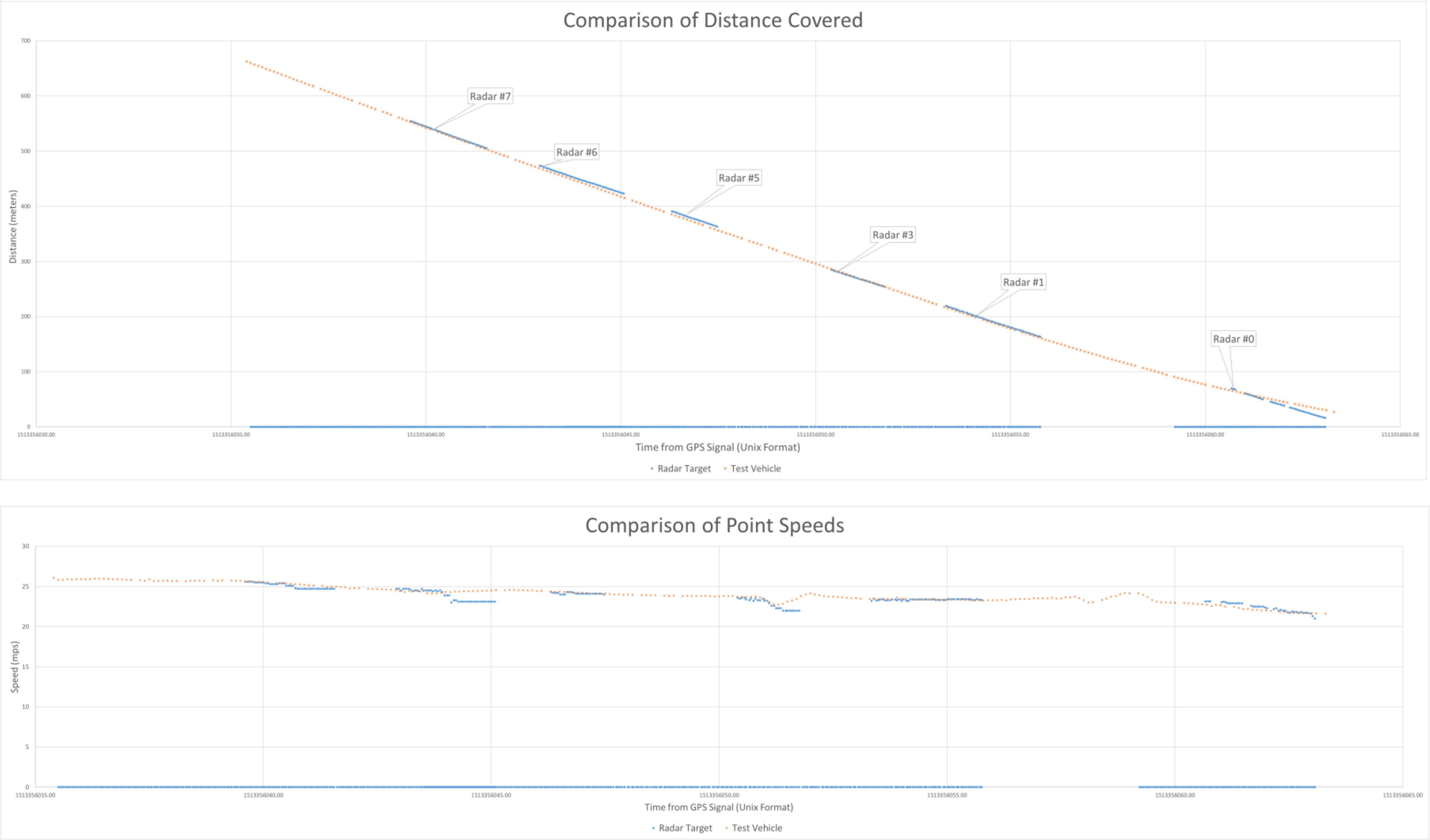


Figure 7-4 Comparison of Radar Measurements with Test Vehicle OBD Position and Speed. Middle Lane (Second Pass)

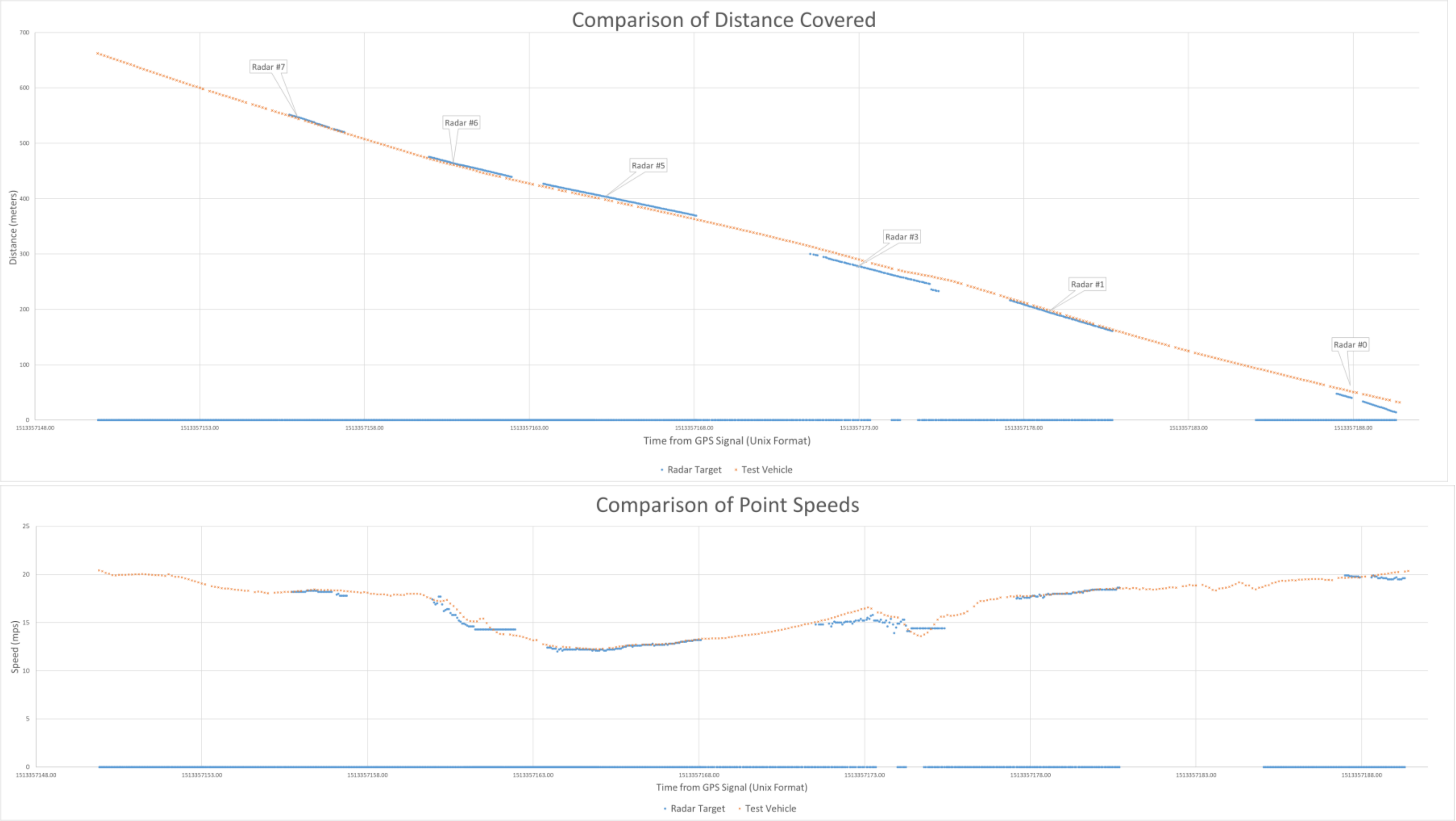


Figure 7-5 Comparison of Radar Measurements with Test Vehicle OBD Position and Speed. Right Lane (Last Pass)

CHAPTER 8: IMPLEMENTATION OF SAFETY APPLICATIONS

Two freeway traffic management applications were targeted as test implementations on the I-94 CV Testbed, the Queue-WARN and Speed-HARM. At the time of this project's conception, it was envisioned that, on separate parallel efforts, CV pilot vehicles will become available as well as human factors projects will utilize the testbed. Unfortunately, no such development materialized, rendering impossible the actual testing of the aforementioned INFLO applications. This is the reason why, although one DSRC OBD and one RSU units were acquired and integrated into the system, they were not permanently deployed in the field. Such deployment would have involved additional infrastructure to be developed upstream of the main testbed area. Tests were conducted with portable stations but lacking actual instrumented vehicles only the hardware/software capabilities were tested. At the same time the effort establishing the CV Testbed was ongoing, a project funded by MnDOT, produced among other a working, real-time, infrastructure based Queue Warning system. The real-time system utilized the original labs machine vision point sensors providing individual vehicle speed measurement in two locations downstream of the Portland Ave overpass. A first test implementation of the CV Testbed instrumentation involved the testing of the Infrastructure based queue warning system with data provided by the CV Testbed radar network. Following, that the basic components of the INFLO applications were tested.

8.1 MIGRATION OF AN INFRASTRUCTURE-BASED QUEUE-WARN TO THE CV TESTBED

Work on the identification and detection of Crash-Prone Conditions was the force behind the original establishment of the I-94 Field Lab in 2002 as part of the Minnesota Traffic Observatory (MTO) of the University of Minnesota. Hourdakis et al. (*2004*) reported details of the site instrumentation and capabilities. As shown in Table 8-1, different types of data were used. Individual vehicle measurements of speed and time headway were used for the operation of the system and aggregated traffic speed data from loop detectors serve for system adjustment and system evaluation. In order to collect the necessary individual vehicle data cost-effectively and unobtrusively, MVDs were utilized (Figure 8-1). Due to the high concentration of conflict events at the test site, only two such sensors stations were necessary, one placed at the location of the most frequent crashes and the second approximately 750 feet downstream. The two stations were deployed between 3rd Ave and MN 65 and between MN 65 and Portland Ave.

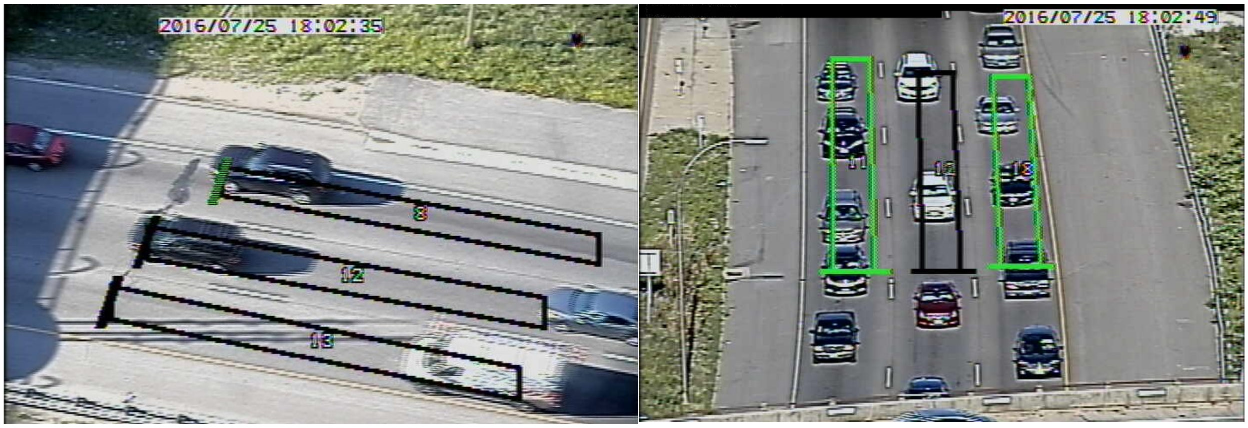


Figure 8-1 Machine Vision Sensors

The project also utilized MnDOT in-pavement loop detectors to provide 30 second volume and occupancy data to provide additional information for the system adjustment from policy makers. Five surveillance cameras were also employed: four atop a high-rise building to capture vehicle conflict events between 3rd Ave and Chicago Ave on video and one atop a pole to capture live video of the MnDOT changeable message boards at the 11th Ave gantry. Video from all five cameras was captured and saved digitally from 9 a.m. to 8 p.m. every day. Vehicle data from the loop detector is collected, 24 hours a day, 7 days a week whereas the individual vehicle measurements were only collected between 7 a.m. and 8 p.m. each day. Traffic event data extracted from these surveillance videos were used to measure the performance of the proposed system in a real-world context.

Table 8-1 Summary of data types and purpose

Data	Type	Source	Purpose
Aggregated Traffic Data	Real-Time	Loop Detector	Additional input for system adjustment
Aggregated Traffic Data	Historical	Loop Detector	Developing of system evaluation methodology
Individual Vehicle Measurements	Real-Time	MVD	The major input of the system
Individual Vehicle Measurements	Historical	MVD	Algorithm and system Design and development
Traffic Event Data	Historical	Video	Developing of system evaluation methodology, algorithm and system design and development

8.2 TRAFFIC MEASUREMENTS AND METRICS

Several metrics were calculated using individual vehicle information such as speed and headway in order to quantitatively describe traffic conditions. As individual vehicle measurements hold the benefit of having much detailed information in high resolution, they also carry large amount of stochastic noise. This fact brings a paradox that aggregation can reduce the impact of noise but also result in loss of detailed information while increase the resolution may bring more noise. Traditionally, individual vehicle measurements are aggregated in time to produce averages. While the aggregated data has less noise, it can no longer describe both the temporal and spatial nature of different traffic flow conditions.

In order to obtain elaborate information without much noise, a multi-metric approach was utilized, which aggregates the data into different traffic metrics. This approach reduces the impact of noise by aggregating individual vehicle measurements over space and time while the combination of different metrics attempts to compensate for the loss of information during the aggregation and quantify more characteristics of the traffic flow. To that end, Hourdos et al. (2005) proposed a series of traffic metrics derived from individual vehicle measurements, both temporal and spatial, to detect crash-prone conditions in freeway traffic. Variations of metrics were also introduced to reflect aggregation over time and space. These metrics include average speed, coefficient of variation of speed, traffic pressure, kinetic energy, coefficient of variation of time headway, coefficient of variation of space headway, acceleration noise, mean velocity gradient, quality of flow index, and a number of heuristic metrics calculated with data from multiple detectors.

Generally, a moving average window approach was utilized in this report to perform the translation from individual vehicle measurements to these metrics. With a sequence of individual vehicle speeds, the entries needed for a specific metric will be selected by the size and time shift of such moving window. Window size represents the number of vehicles in a window. Prior time shift determines the location of the moving window and it decides the time distant between the last vehicle in the window with the concept of “current time” in a real time system. In this report, window sizes were chosen from the set {15,30,40,50,60,70,80,100,110,120} in vehicles and prior time shifts were selected from the set {10,30,60,120,180,240,300} in seconds. The variations in temporal and spatial metrics that follow will be denoted in the form Metric-Location-Lane-Window_size-Window_end_time. For example, AvgSp-Down-R-15-30 denotes the average speed among 15 vehicles on the right lane of the downstream station at least 30 seconds ago.

8.2.1 Temporal Metrics

The following are the definitions of a few of the metrics used in the estimation of the crash probability.

8.2.1.1 Average Speed

Average speed is a common and informative statistic and helps reduce stochastic noise.

8.2.1.2 Coefficient of Variation of Speed (CVS)

In addition to averaging, standard deviation is also a popular way to measure data dispersion. The coefficient of variation, also called relative standard deviation, standardizes the actual standard deviation by its sample mean. The CVS is the product of the standard deviation and the mean value of the speed. As its definition implies a higher value of the coefficient of variation of speed means higher variability in the speed data.

8.2.1.3 Coefficient of Variation of Time Headway (CVTH)

The time headway (TH) between vehicles is an important metric that describes safety and level of service. TH calculation requires individual vehicle arrival times at a point and is simply the difference between the arrival times of two successive vehicles. For the purposes of this research, the actual time headways are not as important as the magnitudes and rates of their change, so the chosen metric is the CV of TH in a group of n vehicles.

8.2.2 Spatial Metrics

8.2.2.1 Density

Density (k) is defined as the number of vehicles per unit length. It is an important characteristic of traffic flow in many models describing its relationship with speed. There are several different models that measure density such as a linear model by Greenshields, a log model by Greenberg, an exponential model by Underwood, and many others. In this report, density is not used directly but rather as a component in the calculations of other traffic metrics such as traffic pressure and kinetic energy.

8.2.2.2 Acceleration Noise

A stochastic continuum theory must be able to quantify the noisy character of traffic flow due to individually different accelerations of the vehicles which build up a regarded ensemble. The drivers of such an ensemble are influenced by many disturbances like bumps, curves, lapses of attention, and different engine capabilities. The acceleration of a regarded vehicle can be split into a term which describes velocity control within a car following model and a random term which is the natural acceleration noise. This noise is usually defined as the root mean square deviation of the acceleration of the vehicle driven independently of other vehicles (Herman et al. 1959). Besides the dependence on the type of road, the number of curves, and the occurrence of bottlenecks in traffic, the acceleration noise is a function of the density and traffic volume. In the original implementation of the Queue Warning system, the calculation of the acceleration noise in this study follows the approximation proposed by Jones and Potts (1962). As will be discussed later, this is one of the metrics that could be calculated directly from the radar measurements.

8.2.2.3 Mean Velocity Gradient

In order to differentiate between different traffic conditions with similar acceleration noise, such as slow, congested traffic versus fast traffic inside a shockwave, Helly and Baker (1965) proposed another measurement, the mean velocity gradient, described by Equation 8.1.

$$MVG = \frac{\sum_{i=1}^{N}(MVG_i)}{N} \qquad (8.1)$$

$$MVG_i = \frac{(AN)_i}{\overline{u}_i}$$

Where

MVG : Average Mean Velocity

MVG_i : Mean Velocity Gradient of vehicle i

N : Total number of vehicles in a hypothetical mile

$(AN)_i$: Acceleration Noise

$\overline{u}_i$: Average speed (mean velocity) of vehicle i

8.2.2.4 Quality of Flow Index

The quality of flow index proposed by Greenshields (1961), provides a quantitative metric to describe the safety of the traffic conditions on a given road based on the number of speed changes and their frequency (Equation 8.2).

$$QFI = \frac{\sum_{i=1}^{N} QFI_i}{N} \qquad (8.2)$$

$$QFI_i = \frac{k\overline{u}}{\Delta u}\sqrt{f}$$

Where

QFI : Average Quality of Flow Index

QFI_i : Quality of Flow Index of Vehicle i

N : Total number of vehicles in a hypothetical mile

$\overline{u}$: Average speed

Δu : Absolute sums of speed changes in a mile

f : Number of speed changes in a mile

k : Constant of 1000 when speed unit is mph and the length of the section is one mile.

8.2.2.5 Traffic Pressure

Traffic pressure (TP) was designed to measure the smoothness of traffic flow. It is defined as the product of speed variance and density (Philips, 1979) as seen in Equation 8.3a. As discussed previously, a higher density is generally associated with a lower average speed. When both the density and the variance of speed are high, it may indicate a “stop-and-go” traffic that could be dangerous and crash prone.

$$TP = \sigma_s^2 \times k \qquad (8.3a)$$

Where

TP: Traffic Pressure

σ_s^2 : Speed variance

k: Density

8.2.2.6 Kinetic Energy (KE)

Kinetic Energy is a familiar quantity in the world of physics that represents the energy of a moving object. This measurement can also be modified to quantify the energy stored in the traffic flow. In the context of traffic flow, kinetic energy measures the energy in the motion of the traffic stream.

Similar to the kinetic energy in physics, according to energy conservation law, within the given traffic system the total amount of energy will not change but can change its form. Drew (1936), described the antithesis of kinetic energy as internal energy that is erratic motion due to geometrics and vehicle interactions and corresponds to an earlier description of Acceleration Noise. Please note that the Kinetic Energy in this study is for traffic flow and it is different from the kinetic energy of a moving object, which means it is not dependent on the mass of vehicles but instead on the density of the traffic stream. The formulation of KE is described in equation 8.3b.

$$KE = ak(u)^2 \qquad (8.3b)$$

Where

a: kinetic energy correction parameter, a dimensionless constant, here is 1

k: density of the traffic stream

$\bar{u}$: average speed of the stream

8.2.3 Heuristic Metrics

8.2.3.1 Up/Down Speed Difference

The up/down speed difference is the difference between the maximum vehicle speed at the upstream sensor and the minimum vehicle speed at the downstream sensor. Its purpose is to measure the travel behavior of a shockwave. For example, when traffic is smooth without shockwaves, the up/down speed difference should be small. When a shockwave has reached the downstream sensor, but has not yet reached the upstream sensor, there should be a lower speed downstream than upstream thus resulting in a high up/down speed difference. A positive Up/Down Speed Difference indicates that the maximum speed of upstream is higher than the minimum speed of downstream. On the other hand, a negative sign of Up/Down speed difference indicates that the maximum speed of upstream is lower than the minimum speed of downstream. The latter case may happen when upstream is in congestion and downstream traffic already recovered from congestion.

8.2.3.2 Right/Middle Lane Speed Difference

As the name suggest, this metric is the difference in speeds between the right lane and the middle lane. When the traffic on the right lane is significantly slower than that on the middle lane, lane changes become more dangerous as they require drivers to divert their attention from the traffic ahead and search for a gap in their mirrors. This increases their reaction time and can be dangerous when shockwaves approach.

8.2.3.3 Max/Min Speed Difference

This metric measures difference between the maximum speed and minimum speed on a sensor location. When a shockwave hits a location, in a relatively short number of vehicles, the speeds tend to fluctuate and drop and results in a high Max/Min Speed Difference. Such high value is usually observed in the occurrence of traffic oscillations and crashes.

8.2.4 Migration to Data from the CV Testbed

The first priority was not to modify the underlying methodology of the detection of Crash Prone Conditions but to improve the estimation of the system parameters. From the aforementioned types of metrics, it is clear that the Temporal metrics would change fundamentally since they are based on measurements on the road over variable time periods. In difference, the Spatial Metrics and primarily Density, are only estimated based on the MV spot sensors, while with the availability of vehicle trajectory data no assumptions are needed and the metrics can be calculated based on direct measurements. For example, Acceleration Noise, as a spatial metric is defined as the standard deviation of the acceleration about the mean acceleration of a single vehicle over a trip of finite distance. Lacking actual spatial information, the MnDOT project assumed that each of the vehicles passing over a point will change their speed to match the speed of their preceding vehicle. Radar collected vehicle trajectories are directly used to calculate Acceleration Noise and in similar way, Mean Velocity Gradient, Quality of Flow, and Kinetic Energy.

Other than the direct calculation of the Spatial metrics, no other changes were introduced on the Infrastructure based Queue Warning system. The biggest benefit from the use of the Radar system was that the algorithm achieved similar performance without requiring to filter that data to remove stochastic noise and autocorrelation effects as was the case with the original system. The filtering of the raw data is a very resource consuming process that imposes limitations of the size of the protected freeway section.

Given that this was a separate funded project, the system working under the CV Testbed was not connected to the field to avoid compromising the evaluation.

8.3 CRASH PROBABILITY MODEL

Given the described metrics and their variants as well as the selected digital filter, a crash probability model was produced. The model is based on a fitted logistic regression model. The experiment in this study applied the model described by Hourdos (2005) to reflect the likelihood of a crash, which is the main input in the crash-prone condition detection algorithm and resulting queue warning system.

8.3.1 System Architecture

This section presents the system architecture. As shown in Figure 8-2, the system follows a three-layer design. The Crash Probability layer collects real-time individual vehicle measurements and processes them to remove noise. The filtered data then pass to the crash-probability model to assess the likelihood of a crash. This crash likelihood along with additional traffic information such as speeds and headways are passed to the second layer, the Algorithm layer. In this layer, the algorithm decides if a warning message should be generated by comparing the crash probability with preset thresholds and real-time traffic conditions. A decision of whether to raise or drop the alarm is being generated and passed to the third layer: System Control, in which requirements from policy makers are applied to modify the result before delivering it to the message sign in the field.

8.3.2 Control Algorithm

This section describes the second layer of the system, the control algorithm. The control algorithm is developed to determine when to start and stop the warning message. The inputs in the algorithm are the crash probability and the filtered vehicle speeds. A moving median filter, or average filter, is applied to the crash probabilities to reduce noise and outliers. A dynamic average window methodology is used to calculate the adjusted crash probability for real-time traffic conditions. Based on this adjusted crash probability, user-defined thresholds, and the result of a speed test, the decision of whether to raise the alarm is made by the system. Once the alarm is raised, it remains active for a minimum of one minute regardless if the crash probability drops below the threshold. This assures that the sign will remain active throughout the trajectory of the shockwave that raised the alarm. Each subsequent alarm renews the one-minute extension.

8.3.3 Control Logic

The crash likelihood alone is not sufficient for determining the precise time to warn the drivers while still producing consistent and stable decisions. Since the probability is a continuous variable, a two-threshold approach is employed to determine whether a crash likelihood indicates crash-prone conditions. One threshold for determining the raising of the alarm and a second for determining its termination. In addition to the crash probability, the algorithm proposed in this report also takes additional traffic measurements into consideration to increase the accuracy and efficiency of the alarm.

Once the alarm is initiated, it will remain active for a minimum time period, currently one minute. Since the individual vehicle data is inherently noisy and can change very quickly, the single model structure used to measure the crash probability can generate spikes when it encounters speed outliers. A noisy spike caused by outliers can be considered as a sudden surge in the value of crash probability and a sudden drop very short time after the surge. The values before and after such spike were usually in similar ranges. Therefore, such spike may activate the alarm trigger for only several seconds where the alarm shouldn't be raised. Thus, deciding the result of the alarm only by the alarm trigger can lead to frequent changing of warning messages. The control logic of the algorithm can be found below in Equation 8.4.

$$\begin{aligned}
&Greater(\overline{p_t},\lambda_1) \wedge SpeedTest(v_{tb}) \rightarrow AlarmOn \\
&AlarmOn \wedge Greater(\lambda_2,\overline{p_t},) \wedge TimeCheck(t) \rightarrow AlarmOff \\
&SpeedTest(v_{tb}) = \begin{cases} true & v_{tb} \leq u_1 \\ false & v_{tb} > u_1 \end{cases} \\
&TimeCheck(t) = \begin{cases} true & t - t_0 > \Delta t \\ false & t - t_0 \leq \Delta t \end{cases}
\end{aligned} \tag{8.4}$$

Where

u_1 : test speed, default is 45 mph. (mph)

t_0 : last time in the past with an alarm trigger (s)

λ_1 : Starting Threshold

λ_2 : Ending Threshold

v_{tb} : Current Downstream Speed (mph)

$\overline{p_t}$: Adjusted Crash Probability

$$\overline{p_t} = \frac{1}{n} \sum_{i=t-n+1}^{t} p_i \qquad n = \begin{cases} f(\overline{p_t},\lambda_1) & \overline{p_t} \geq \lambda_1 \\ g(\overline{p_t},\lambda_1,\lambda_2) & \lambda_2 \leq \overline{p_t} < \lambda_1 \\ h(\overline{p_t}) & \overline{p_t} < \lambda_2 \end{cases} \tag{8.5}$$

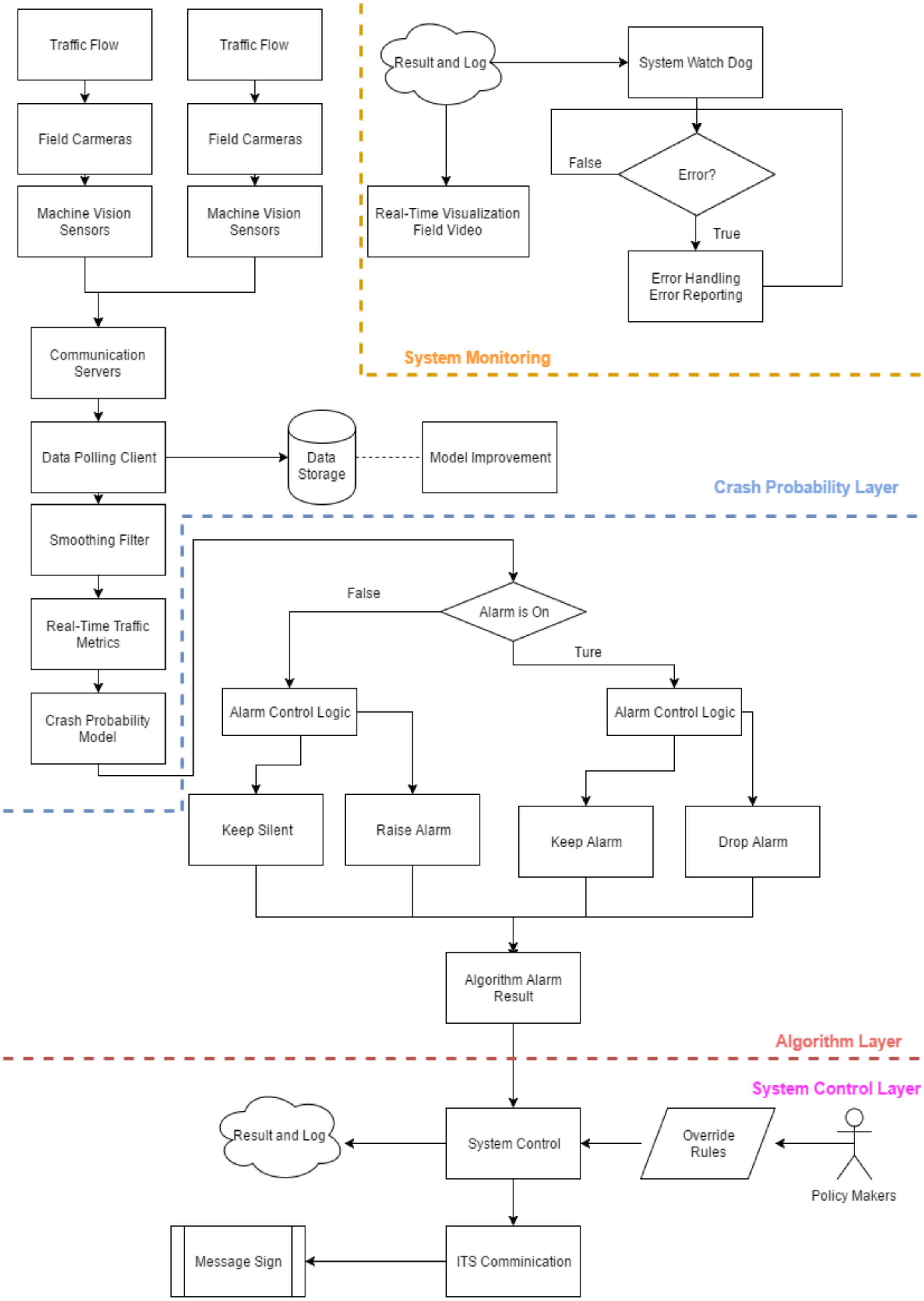


Figure 8-2 Infrastructure Queue Warning System Architecture

8.4 TESTING OF INFLO QUEUE-WARN AND SPEED-HARM

The objective of Q-WARN is to provide a vehicle operator with sufficient warning of an impending queue backup in order to brake safely, change lanes, or modify the route such that secondary collisions can be minimized or even eliminated. It is distinct from collision warning, which pertains to events or conditions that require immediate or emergency actions. Queue warnings are provided in order to reduce the likelihood of the formation of such emergency events.

A queue backup can occur due to a number of conditions, including:

- Daily recurring congestion caused by bottlenecks
- Work zones, which typically cause bottlenecks
- Incidents, which, depending on traffic flow, lead to bottlenecks
- Weather conditions, including icing, low visibility, sun angles, and high wind
- Exit ramp spillovers onto freeways due to surface street traffic conditions

In all cases, queuing is a result of significant downstream speed reductions or stopped traffic and can occur with freeways, arterials, and rural roads. Queuing conditions present significant safety concerns; in particular, the increased potential for rear-end collisions. They also present disruptions to traffic throughput by introducing shockwaves into the upstream traffic flow. A queue warning system will be successful at minimizing secondary collisions and the resulting traffic flow shockwaves by being able to: rapidly detect the location, duration, and length of a queue propagation; formulate an appropriate response plan for approaching vehicles; and disseminate such information to the approaching vehicles readily and in an actionable manner.

The INFLO Q-WARN application concept aims to minimize the occurrence and impact of traffic queues by using connected vehicle technologies, including vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communications, to enable vehicles within the queue event to automatically broadcast their queued status information (e.g., rapid deceleration, disabled status, lane location) to nearby upstream vehicles and to infrastructure-based central entities (such as the TMC). The conceptual Q-WARN application performs two essential tasks: queue determination (detection and/or prediction) and queue information dissemination. In order to perform these tasks, Q-WARN solutions can be vehicle-based or infrastructure-based or utilize a combination of each.

It is important to note that the Q-WARN application concept is not intended to operate as a crash avoidance system (e.g., like the forward collision warning [FCW] safety application). In contrast to such systems, Q-WARN will engage well in advance of any potential crash situation, providing messages and information to the driver in order to minimize the likelihood of his/her needing to take crash avoidance or mitigation actions later. As such, Q-WARN-related driver communication will always give priority to crash avoidance/ mitigation safety applications when such applications determine that a safety-related warning is necessary.

Dynamic Speed Harmonization (SPD-HARM): The objective of SPD-HARM is to dynamically adjust and coordinate maximum appropriate vehicle speeds in response to downstream congestion, incidents, and weather or road conditions in order to maximize traffic throughput and reduce crashes. A dynamic SPD-HARM system will be successful at managing upstream traffic flow by being able to: reliably detect the location, type, and intensity of downstream congestion (or other relevant) conditions; formulate an appropriate response plan (i.e., vehicle speed and/or lane recommendations) for approaching vehicles; and disseminate such information to upstream vehicles readily and in a manner which achieves an effective rate of compliance. Improved safety results, in terms of reduced crash rates and less severe crashes, have shown to be the most significant and consistent achievements across deployments that exist today at some level. In addition, SPD-HARM techniques promote reduced vehicle speeds and speed variance, especially in unsafe driving conditions; support modest improvements in throughput; and have a moderately positive impact on travel time reliability. There are three key factors that contribute to the operation of an effective speed harmonization system. The first factor is the availability of information describing the prevailing traffic conditions. The second factor is the availability of a reliable strategy (algorithm) for selecting speed limits. The last factor is the flow of information between the field and the decision making center or the location where the Speed Harmonization control algorithm resides.

Research and experimental evidence has consistently demonstrated that by reducing speed variability among vehicles, especially in near-onset flow breakdown conditions, traffic throughput is improved, flow breakdown formation is delayed or even eliminated, and collisions and severity of collisions are reduced. The INFLO SPD-HARM application concept aims to realize these benefits by utilizing connected vehicle V2V and V2I communication to detect the precipitating roadway or congestion conditions that might necessitate speed harmonization, to generate the appropriate response plans and speed recommendation strategies for upstream traffic, and to broadcast such recommendations to the affected vehicles.

Because the INFLO applications are so closely linked, the effectiveness of each can be improved by taking advantage of the benefits to traffic flow and safety that the others provide. In fact, research-to-date has shown that the most successful implementations have been those that combine multiple different freeway management control applications. For example, SPD-HARM benefits Q-WARN by slowing and managing upstream traffic, thus reducing the risk of secondary collisions. CACC benefits SPD-HARM by providing a mechanism for harmonizing traffic flow and reducing or mitigating acceleration variability. Q-WARN benefits CACC by providing the platoon sufficient notification of an impending queue to effectively manage a response.

Importantly, SPD-HARM and Q-WARN are technologies that can be implemented in the near-term. Their benefits are optimized when implemented as infrastructure-based applications that reside at a central entity such as a Traffic Management Center (TMC) as the TMC system has broader visibility into the traffic state, allowing operators to implement a more proactive approach for predicting queues and congestion.

In addition to the benefits of deploying the two bundled INFLO mobility applications in concert, the applications would also benefit from integrating with other applications, including safety systems like electronic stability control (ESC) systems, night vision systems, curve speed warning systems, lane departure warning systems, alcohol monitoring systems, brake assist systems, steering assist systems, forward collision warning (FCW) systems, and pre-crash sensing systems. Coordination with ramp metering systems would also help provide the INFLO applications a better connection with the overall transportation network. Finally, integrating the INFLO applications with Advanced Traveler Information Systems (ATIS) would provide road users enhanced information about the state of the transportation system, pre-trip planning, route-making, and incident avoidance.

8.4.1 Integration of INFLO in the I-94 CV Testbed

The integration of the two bundled INFLO applications followed the small scale demonstration performed by Battelle in Seattle. In that demonstration, given that only 3 or 4 connected vehicles were available, the detection of the queue was based on infrastructure sensors as well as vehicle produced BSM communicated via cellular links. The directions to the upstream CVs was restricted in the generation of messages informing the driver of his/her distance to the back of the queue. In the Seattle demo the back of the queue was estimated based on the last reported location but in the case of the I-94 CV Testbed, given the continuous coverage with the radar sensors, the location of the last vehicle was reported as detected. For purely demonstration purposes the information was transmitted via SMS text message to the cellphone of the approaching driver. This was a proof of concept only since no instrumented vehicles were available nor the project has an available funding to support such deployment.

CHAPTER 9: CONCLUSIONS AND FUTURE DEVELOPMENTS

Safety and traffic operations concepts based on V2I and V2V communication have been in development for some time. Since 2003, with the assistance of the Minnesota Department of Transportation, the Minnesota Traffic Observatory at the University of Minnesota has studied and experimented with infrastructure-based Q-WARN systems. A permanent field lab has been established at the high-crash area of westbound I-94 in Minneapolis, capturing detailed data on hundreds of crashes. This area experiences upward of 100 crashes annually, the majority of them rear-end crashes due to failure to stop or too little headway. This research capitalized on the already extensive instrumentation available at the I-94 Field Lab to develop a CV testbed specifically for the implementation and testing of SPD-HARM and Q-WARN systems. The current site was enhanced to support fully developed CV safety systems as well as the research and evaluation of the underlying human factors of such systems. The final product of this project was a fully functional CV testbed uniquely situated to attract freeway safety-oriented V2I and V2V safety application development, implementation, and evaluation projects.

Although the original plan was to deploy overlapping radar sensors on a half-mile stretch of I-94, following several meetings and discussions with MnDOT, it became clear that the agency would not allow the MTO project team to install the sensors on a temporary basis and that MnDOT required a secure permeant deployment specifically in terms of the power-line conduits. The project, as budgeted, did not have the funds for such a deployment. Instead, as much as possible the project utilized existing MnDOT camera poles even when their location limited coverage and, in some cases, accuracy of the sensors. The reason for this decision was that, at the present stage, the priority has been to establish the backbone of the sensor communication network and data collection system.

In addition to the aforementioned issues, with the radar station deployment, an additional problem involving the radar was encountered. The project had proposed using the latest generation of radar sensors, which provides higher performance and accuracy in heavy traffic sites like I-94. The proposal was developed following several reassurances from the manufacturer that these latest models would be available in the US shortly after the start of the project. The project officially started in November 2014, and the agreement was that we would purchase the latest generation, but until this generation was available and allowed to be sold in the US, we would receive the older generation so the project could proceed with the initial testing and development. In February 2015 after considerable pressure on the manufacturer, it revealed it did not anticipate making the sensors available until February 2016 because the matter was caught up in litigation in US courts. Given this reality, the project team decided to proceed with the older generation stations, develop and deploy the system, identify where the older generation sensors were insufficient and by how much and plan for a later upgrade to better ones.

REFERENCES

Balke, K., Charara, H., & Sunkari, S. (2014). *Report on Dynamic Speed Harmonization and Queue Warning Algorithm Design.* Washington, DC: United States Dept. of Transportation.

Davis, G., & Hourdos, J. (2012). *Development of Analysis Methods Using Recent Data.* Washington, DC Transportation Research Board.

Dirks, P., Liu, Z., & Hourdos, J. (2016). Freeway Traffic Queue-Warning System: Deployment and First Impressions. *Transportation Research Board 96th Annual Meeting.* Washington DC: Transportation Research Board.

FCC. (2013). Retrieved July 18, 2017, from https://apps.fcc.gov/edocs_public/attachmatch/FCC-13-98A1_Rcd.pdf, "FCC 13-98: Report and Order and Further Notice of Proposed Rulemaking"

Federal Communications Commission. (2015). *Operation of Radar Services in the 76-81 GHz Band.* Washington, DC: Federal Communications Commission.

Greenshields, B. D. (1961) Quality of Traffic Flow, Quality and Theory of Traffic Flow (pp.3-40). Symposium, Bureau of Highway Traffic, Yale University, New Haven, CT.

Helly, W., & Baker, P. G., (1965). Acceleration noise in a congested signalized environment. Vehicular Traffic Science (pp. 55-61). Proceedings of the Third International Symposium on the Theory of Traffic Flow, American Elsevier, New York.

Hill, C., & Krueger, G. (2015). *ITS ePrimer Module 13: Connected Vehicles*. Retrieved November 16, 2017, from https://www.pcb.its.dot.gov/eprimer/module13.aspx

Hourdakis, J., Michalopuloulos, P., & Morris, T. (2004). Deployment of Wireless Mobile Detection and Surveillance for Data-Intensive Applications. *Transportation Research Record*, *1900,* 140–148.

Hourdakis, J., Morris, T., Michalopoulos, P., & Wood, K. (2005). *Advanced Portable Wireless Measurement and Observation Station.* Minneapolis, MN: Intelligent transportation Systems Institute, Center for Transportation Studies, University of Minnesota.

Hourdos, J. (2005) Crash prone traffic flow dynamics: Identification and real-time detection. Ph.D. dissertation (315 p.), University of Minnesota, Minneapolis, MN.

Hourdos, J., & Zitzow, S. (2014). *Investigation of the Impact of the I-94 ATM System on the Safety of the I-94 Commons High Crash Area.* St. Paul, MN: Minnesota Department of Transportation Research Services & Library.

Hourdos, J., Garg, V., & Michalopoulos, P. (2008). *Accident Prevention Based on Automatic Detection of Accident Prone Traffic Conditions: Phase I.* Minneapolis, MN: Intelligent Transportation Systems Institute, University of Minnesota.

Hourdos, J., Liu, Z., Dirks, P., Liu, H. X., Huang, S., Sun, W., & Xiao, L. (2017). *Development of a Queue Warning System Utilizing ATM Infrastructure System Development and Field-Testing.* St. Paul, MN: Minnesota Local Road Research Board, Minnesota Department of Transportation Research Services & Library.

Hourdos, J., Xin, W., & Michalopoulos, P. (2008). *Development of Real-Time Traffic Adaptive Crash Reduction Measures for the Westbound I-94/35W Commons Section.* Minneapolis, MN: Intelligent Transportation Systems Institute, University of Minnesota.

Jones, Trevor R., & Potts Renfrey, B. (1962). The Measurement of Acceleration Noise-A Traffic Parameter. *Operations Research, 10*(6), 745–763.

Parikh, G., & Houdos, J. (2014). *Implementation of High Accuracy Radar Detectors for Traffic Safety Countermeasure Evaluation.* Minneapolis, MN: Center for Transportation Studies, University of Minnesota.

Phillips, W. F. (1979). A kinetic model for traffic flow with continuum implications. *Transportation Planning and Technology, 5*, 131–138.

Rindels, M., Zitzow, S., & Hourdos, J. (2016). *Evaluation of the Effectiveness of ATM Messages Used During Incidents.* St. Paul, MN: Minnesota Department of Transportation Research Services & Library.

Smart Microwave Sensors GmbH. (2017, November 9). *UMRR Traffic Sensor Type 29 Data Sheet*. Retrieved from http://www.smartmicro.de/fileadmin/user_upload/Documents/TrafficRadar/UMRR_Traffic_Sensor_Type_29_Data_Sheet.pdf

Smart Microwave Sensors GmbH. (2017, November 9). *UMRR Traffic Sensor Type 30 Data Sheet*. Retrieved from http://www.smartmicro.de/fileadmin/user_upload/Documents/TrafficRadar/UMRR_Traffic_Sensor_Type_30_Data_Sheet.pdf

Xin, W., Hourdos, J., & MIchalopoulos, P. (2008). *Enhanced Micro-Simulation Models for Accurate Safety Assessment of Traffic Management ITS Solutions.* Minneapolis, MN: Intelligent Transportation Systems Institute, University of Minnesota.