

**Design, Specification, Implementation and Evaluation of a  
Real-Time Queue Warning System**

**A THESIS  
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL  
OF THE UNIVERSITY OF MINNESOTA  
BY**

**Zhejun Liu**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
MASTER OF SCIENCE**

**December, 2017**

© Zhejun Liu 2017  
ALL RIGHTS RESERVED

# Acknowledgements

I would like to thank my advisor and a very good friend of mine, Dr. John Hourdos for his guidance and inspirations through my research and study. He is always there for me when I am facing difficulties and challenges in both study and research. This work would not be possible without his encouragement and enlightenment.

I would also like to thank Gordon Parikh who provides a great help in the system developing and testing, many undergraduate students who spend lots of time watching the video to collected traffic event data and Peter Dirks for his help in supervising the video data extraction and evaluation process.

Special thanks to professor Gary Davis and professor Saif Benjaafar for serving on my committee. Thanks to the faculty members in Civil Engineering, Industrial Engineering and Computer Science for providing me great knowledge through wonderful courses.

## Abstract

The formation and propagation of queues on urban freeways is an unavoidable result of the ever-increasing traffic demand. Traffic queues increase on-road congestions and reduce roadway network efficiency. More importantly, some of these queues can cause serious rear-end collisions, resulting in property damages and injuries. This thesis presents the design, specification, implementation and evaluation of an infrastructure based Queue Warning system (QWarn) that is capable of detecting dangerous traffic conditions, i.e. crash-prone conditions, on freeways and delivering warning messages to drivers, in order to increase their alertness and ultimately reduce the frequency of crashes. This effort approaches the topic from the quantification of traffic flow to the multi-layer system design along with different methodologies including the traffic assessment modeling and the development of control algorithms. This study utilizes measurements of individual vehicle speeds and time headways at two fixed locations on the freeway mainline, as the major type of data for the systems operation. This thesis focuses on a case study and evaluation of the proposed system implemented on a high crash frequency freeway section. The Queue Warning system was implemented at the right lane of a 1.7-mile-long freeway segment of Interstate 94 Westbound near downtown Minneapolis where the event frequency prior to the systems installation was 212.25 conflict events per million vehicles traveled. Machine Vision Detectors (MVD) are installed on a nearby rooftop capturing the Real-Time vehicle data. The data were delivered over the Minnesota Traffic Observatorys (MTO) communication network to a server running the main control algorithm. The control algorithm assesses the dangerousness of the given traffic condition and responds with a warning result based on a multi-metrics traffic evaluation model and complex control reasoning that ensures consistency and accuracy. The system translates the warning result into readable messages and delivers them to the two sets of signs located upstream of the detection zone. A three-month investigation of the operations of the QWarn system shown a reduction in conflict event frequency to 135.79 per million vehicle traveled. In conclusion, this thesis proposed the framework as well as details of a Queue Warning system is also evaluated the methodology by implementing a working prototype that delivers warning messages to road



users on an urban freeway segment. The result shows a decrease of conflict event frequency and proves the feasibility of the proposed methodology.

# Contents

<b>Acknowledgements</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Figures</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Thesis Organization . . . . .	3
<b>2 Background</b>	<b>4</b>
<b>3 Study Site and Available Data</b>	<b>8</b>
3.1 Overview . . . . .	8
3.2 Datasets for Off-line Methods . . . . .	14
3.2.1 I94QW2016 Dataset . . . . .	14
3.2.2 MnFW2016 Dataset . . . . .	15
3.3 Data Preprocessing . . . . .	15
3.3.1 Observed and Potential Data Issues . . . . .	15
3.3.2 Duplicated Data . . . . .	15
3.3.3 Missing Data . . . . .	16
3.3.4 Temporal Alignment . . . . .	17

<b>4</b>	<b>Traffic Measurements and Metrics</b>	<b>19</b>
4.1	Temporal Metrics . . . . .	20
4.1.1	Average Speed . . . . .	20
4.1.2	Coefficient of Variation of Speed (CVS) . . . . .	21
4.1.3	Coefficient of Variation of Time Headway (CVTH) . . . . .	21
4.2	Spatial Metrics . . . . .	22
4.2.1	Density . . . . .	22
4.2.2	Acceleration Noise . . . . .	23
4.2.3	Mean Velocity Gradient . . . . .	24
4.2.4	Quality of Flow Index . . . . .	24
4.2.5	Traffic Pressure(TP) . . . . .	25
4.2.6	Kinetic Energy (KE) . . . . .	26
4.3	Heuristic Metrics . . . . .	26
4.3.1	Up/Down Speed Difference . . . . .	26
4.3.2	Right/Middle Lane Speed Difference . . . . .	27
4.3.3	Max/Min Speed Difference . . . . .	27
4.3.4	Individual Vehicle Speed Noise Reduction . . . . .	27
4.4	Extension in Space and Time . . . . .	29
<b>5</b>	<b>System Design</b>	<b>31</b>
5.1	Multi-layer Design . . . . .	31
5.2	State Machine Formulation . . . . .	34
5.3	Dangerous Level Estimation . . . . .	36
5.4	Queue Warning Algorithm . . . . .	36
5.5	System Control . . . . .	40
5.5.1	Policy Adoption: Override Rules . . . . .	41
5.5.2	System Monitoring . . . . .	42
<b>6</b>	<b>System Implementation</b>	<b>43</b>
6.1	Dangerous Level Estimation: A Crash Probability Model . . . . .	43
6.2	Details of the Implemented Prototype . . . . .	44
6.3	System Oversight And Policy Adoption . . . . .	46

6.3.1	External Controls . . . . .	46
6.3.2	Interface . . . . .	46
<b>7</b>	<b>System Evaluation</b>	<b>50</b>
7.1	Traditional Evaluating Method . . . . .	50
7.1.1	Limitations . . . . .	51
7.2	Performance Evaluating Indexes . . . . .	52
7.2.1	Event Type and Notation . . . . .	52
7.2.2	Detection Rate . . . . .	53
7.2.3	Alarm Length . . . . .	54
7.3	The Baseline: A Naive Method . . . . .	54
7.4	Evaluation Details . . . . .	55
7.5	Performance Comparison . . . . .	55
<b>8</b>	<b>Result and Discussion</b>	<b>58</b>
8.1	Detection Rates . . . . .	59
8.2	System Limitations . . . . .	64
<b>9</b>	<b>Conclusion and Future Studies</b>	<b>66</b>
9.1	Conclusion . . . . .	66
9.2	Future Studies . . . . .	67
	<b>References</b>	<b>69</b>

# List of Tables

3.1	Summary of data types and purpose . . . . .	11
6.1	Crash Probability Model [4] . . . . .	44
8.1	Breakdown of system performance in right lane by component . . . . .	60
8.2	Detection rates during the trial period . . . . .	61
8.3	Crash frequencies per million vehicles . . . . .	61
8.4	Near crash frequencies per million vehicles . . . . .	62
8.5	Conflict event frequencies per million vehicles . . . . .	62
8.6	Event frequency comparison . . . . .	62

# List of Figures

3.1	Aerial view of implementation site. . . . .	9
3.2	Machine Vision Sensors . . . . .	12
3.3	Coverage of the four cameras located on roof tops . . . . .	12
3.4	Views of the three essential cameras located on roof tops . . . . .	13
3.5	The view of the sign-monitoring camera . . . . .	13
4.1	Linear Interpolation . . . . .	28
4.2	Spline Interpolation . . . . .	29
5.1	System Architecture . . . . .	33
5.2	State diagram of the system . . . . .	35
5.3	Flowchart of the Queue Warning Algorithm . . . . .	40
6.1	MnDOT changeable message boards with warning displayed. . . . .	47
6.2	Back-end System Interface . . . . .	48
6.3	Back-end System Interface . . . . .	48
6.4	System Simplified Class Diagram . . . . .	49
7.1	Algorithm level evaluation curve(with no override rules) . . . . .	56
8.1	Occupancy Comparison . . . . .	64

# Chapter 1

## Introduction

### 1.1 Introduction

Transportation, especially passenger transportation, has been crucial from the very beginning of human civilization. It shapes the travel behavior of people and has a huge impact on urban development. With the progress of the modern transportation technology, people enjoy the convenience of door-to-door (D2D) travel in a short time, which, however, introduces extra dangers into our daily life. In addition, the rise of automobiles makes the roadways hit the capacity limit. More efficient traffic operations are necessary to ease the tensions. Running a facility close to capacity increases the frequency of instabilities and flow breakdowns also produce dangerous driving conditions.

In 2014, US officials reported the occurrence of 6.1 million crashes. These crashes injured 2.3 million people and killed 32,675 people [1]. In 2014, the Minnesota Department of Public Safety (DPS) reported 15,648 crashes occurred on freeway. These crashes accounted for the death of 38 people and the injury of 5,031 people [3]. Crashes are economically expensive both in money and time. According to the Minnesota Department of Transportation (MnDOT), the average cost of a single crash is \$7,600 when it involves only property damage, \$83,000 to \$57,000 when it involves injury, and as high as \$10.6million when the crash is fatal[2]. Besides, crashes introduce disturbance in the transportation network. Not only delaying the involved travelers, crashes also cause congestion (delaying

others) and reduce the capacities which leads to longer queues and higher delays. Dangerous conditions that lead to crashes bring potential threat to both drivers and pedestrians. They can also create new dangerous conditions due to secondary crashes.

It is essential for people to find a way to reduce such danger brought by freeway transportations and increase the efficiency of freeways. It will not be realistic to stop traveling using the freeway transportation system in this era since it brings much more benefit than danger. Therefore, the goal is to reduce frequency of conflict events so that traveling on freeway can be safer and more efficient. To achieve this, one needs to reveal traffic conditions that are correlated with freeway crashes and the reason why the crash happened in the first place. In a Ph.D. dissertation, Hourdos [28] proposed the causality analysis of freeway rear-end crashes and discovered that many crashes and near-misses are the results of the combination of both flow breakdown and human factors such as dangerous driving or driver distraction. If one can detect or predict the dangerous flow breakdown and provide driver warnings, it is possible to eliminate the human factor in freeway crashes and keep these crashes from happening. Therefore, it is crucial to find the dangerous traffic conditions that serves as one of the factors that leads to crashes.

Traffic shockwaves are transitions between two different traffic conditions. The shockwaves from low density to high density are called compression waves. Compression waves are usually the conditions that queues are built, which related to rear end collisions of vehicles. For simplicity, the rest of this thesis will use the term shockwave to refer compression waves. These shockwaves can be the product of heavy traffic, frequent merging, reduced capacity, and many other factors. When a shockwave is generated, upstream traffic, which is still moving at higher speeds, must decelerate at a high rate. Distracted drivers, an increasing safety concern, could fail to react, causing rear-end collisions, congestion, and increasing the risk of secondary crashes. Although rear-end crashes can happen at any time during a disturbance, research has shown [4] that there are certain traffic conditions that can be reliably associated with crashes. Such traffic conditions are considered as crash-prone. Crash-prone conditions (CPCs) are hard to identify since not all shockwaves carry enough momentum to cause a crash while other localized conditions can also be part



of the causal factors that tip the scale. For example, a shockwave could be triggered by an aggressive lane change that generates a small speed discontinuity, which increases in intensity while traveling upstream. To measure the crash potential of traffic conditions and identify CPCs, traffic measurements with information shifted both in time and space are required. Based on a previous study [4], this thesis proposes a methodology that is capable of generating consistent and discrete warning messages to drivers based on relatively continuous measurement of crash potential and presents a realtime Queue Warning system deployed on a U.S. freeway. The results show that the Queue Warning system successfully reduced frequency of conflict events.

## 1.2 Thesis Organization

A literature review can be found in Chapter 2. Chapter 3 introduces the description of the study site and data collection along with some challenges about real-time data collection. The representation of traffic including traffic measurement and metrics can be found in Chapter 4. Chapter 5 provides an overview of the the multi-layer design of the proposed robust real-time queue warning system. Chapter 6 shows the implementation of the system working on I-94, including its parameters and setup. The author presents an evaluation by comparing the proposed system with a naive algorithm in Chapter 7. Performance comparison among different parameters are also presented in this chapter. The results of the proposed system can be found in Chapter 8. Finally, this thesis concludes itself with a discussion and conclusion in Chapter 9.

## Chapter 2

# Background

Instead of classifying traffic conditions, most queue-warning methodologies only detect the existence of a queue or even the existence of standing queues where traffic is stopped and provide unconditional queue warning accordingly regardless of whether the traffic condition is dangerous or not. A common application of dynamic queue-warning systems is on work zones where reduced capacity of roadways cannot accommodate normal traffic volumes. Two systems were designed for this application, which employed several sensors along the road upstream of the construction zone to detect the location of the queue end. The first queue-warning system [17] used live remote traffic microwave sensor (RTMS) data from two sensors (one was immediately on the upstream of the work area and the other was at the end of the work zone) to extrapolate the location of the back of the queue. The location of the back of the queue was then displayed on a portable changeable message sign (PCMS) at the location of the upstream sensor. The second queue-warning system [18] used up to eight speed sensors spread out over seven miles to detect queues. The system could warn drivers approaching the work zone of any queues present through several upstream PCMSs. Texas Department of Transportation used the eight-sensor system in a 96-mile project and found it lowered the crash rates by as much as 47% when compared to an estimate of what they would have been had the system not been deployed.

For metropolitan areas where congestion is commonplace, different approaches are applied. One system in Houston, TX [19] used MVDs to measure the speeds of vehicles

approaching the area in which congestion generally occurred. When the system detected three consecutive vehicles with speeds lower than 25 mph, lights would flash on a warning sign above. Pesti et al. reported decreases of 2% to 6% in vehicle conflicts when this queue warning system was in place. Another queue warning system was deployed on a congestion-prone freeway in Eugene, OR [20]. The system measured freeway occupancy using three sidefire, dual-beam traffic detectors. When the freeway vehicle occupancy exceeded thresholds established by the Oregon Department of Transportation (ODOT), warnings were displayed on the Portable Changeable Message Signs (PCMSs) until the occupancy no longer exceeded the thresholds along with a minimum display time of 5 minutes. The system was reported to have decreased the crash rate by over 35%.

While the aforementioned systems have been shown to increase driver safety, most of them are operated efficiently only in the presence of standing queues, in which vehicles are not moving or are moving together at very low speeds. As unconditional queue warning systems, i.e. reacting to queues regardless of their dangerousness, these methodologies usually provide warning messages in an unselective manner. In a highly volatile traffic environment, the forming of the queue, or shockwave, can be very short and alter the traffic conditions blisteringly. Existing methodologies tend to have inconsistent results owing to the unstable nature of traffic characteristics under the impact of shockwaves. In addition, they may have a lower efficiency when congestion hits the freeway, since congestion has very similar patterns as normal queues. This study refers these systems as naive systems since they focus on the binary existence of a queue instead of warning drivers about dangerous traffic conditions.

The United States Department of Transportation (USDOT) is currently developing a connected-vehicle traffic management system called Intelligent Network Flow Optimization (INFLO) incorporating queue-warning components (It has not been deployed to date) [21]. The system will rely on vehicle-to-infrastructure(V2I) communication to share traffic data for the purpose of warning queues, harmonizing speed limits, and coordinating cruise control speeds of platooning vehicles. But since V2I has not been applied on most of the vehicles on road, such system may not be operational for the general traffic in the near

future.

Instead of warning people unconditional based on the existence of queues, a conditional warning methodology is desired that can classify traffic conditions. This can be achieved by analyzing the correlation between certain traffic conditions and freeway crashes. In the literature, there are efforts into finding the factors that are highly related to crashes. Discerning what factors influence crash rates and how they do so has been the aim of several research projects. Qiu and Nixon [5] reviewed the effects of weather on the likelihood of a crash. Kopelias et al. [6] studied how the combined effects of geometric, operational, and weather factors influence crash frequencies.

Traffic on urban freeways, however, tends to react differently to those factors. Golob and Recker [7] found that although collisions with objects and crashes of multiple vehicles are more frequent on wet roads, rear-end crashes are more likely to occur on dry roads with good visibility. They proposed that these rear-end crashes were highly correlated to large differences in individual vehicle speed like those seen in stop-and-go traffic. Such traffic conditions are often referred to as traffic shockwaves., which are considered as a type of traffic oscillations. Zheng et al. [8] studied the effects of traffic oscillations on freeway crashes where they found that traffic oscillations and congestion were highly correlated to freeway crashes.

Various crash-prediction models [9–14] have been developed using those factors, and significant association has been found between crash probability and specific traffic conditions. Hourdos et al. [4] utilized crash probability to detect Crash Prone Conditions(CPCs) on a freeway in Minneapolis, MN. Different approaches to apply crash-prediction models in real-time environments have been proposed as well [14–16], however, implementation of such models into real-time systems is rare. This could be caused by the practical limitations, such as real-time data collection and computational issues, for those highly theoretical methodologies.

In the purpose of consistency, accuracy, and efficiency, a conditional and practical queue warning methodology is expected. Such a warning system can react to shockwaves in a

selective manner and only warn drivers when the shockwave is dangerous and crash-prone. Such a system should also be able to terminate the alerting state when traffic recovers to a safer condition. In the purpose of conflict-event frequency reduction and fill the gap between theoretical works and practical implementable systems, this thesis proposed the design, specification, implementation and evaluation of a Queue Warning (QWarn) system that is capable of detecting dangerous traffic conditions, i.e. crash-prone conditions, on freeways and delivering warning messages to drivers, in order to increase their alertness to these traffic conditions and ultimately reduce the conflict-event frequency on urban freeways. This system is robust, efferent and was deployed on a real-world freeway segment(I-94 Westbound) delivering warning messages to drivers in Minneapolis, MN.

This study used individual vehicle measurements of speeds and time headways as the major type of data for the system operation. Based on the traffic metrics and crash probability model proposed by Hourdos [4], this thesis started with the quantification of the traffic flow and the measure of dangerous level. Then a multi-layer design that both separates and connects the dangerous level measurement, algorithm suggestion, system control and policy adaptation was proposed. Detailed design, specification and the implementation of a prototype were fully explained in this thesis as well.

## Chapter 3

# Study Site and Available Data

### 3.1 Overview

To best study crashes, conditions must be observed and measured before and during an actual event. This requirement translates to a need for continuous monitoring and data collection at a location that maximizes the probability of capturing crashes on tape.

In the Minneapolis-Saint Paul Metropolitan Area (Twin Cities), Interstate 94 (I-94) connects these two major cities, and it is the major freeway corridor connecting the two banks of the Mississippi River in the region. The site of this study is a segment with a length of 1.7 miles on I-94 westbound near downtown Minneapolis. This segment of I-94 is near the exit of I-35W, the west branch of Interstate 35 that goes through Minneapolis. Being near the downtown results in a large traffic volume merging in and out these two freeways. Such a large volume and merging often destabilizes traffic on this freeway segment, resulting in rear-end crashes. A detailed causality analysis of the crash on this site can be found in a previous work by Hourdos [28]. These dangerous traffic conditions represent shockwaves where drivers need to reduce their speed with a short reaction time. According to MnDOT records, this segment had the highest crash rate in the entire freeway system, as high as 4.81 crashes per million vehicle miles. An equivalent to approximately a crash every two days. Fatalities and severe injuries are very rare since the prevailing speed during CPCs is relatively low. The majority of crashes result only in property damage.



Figure 3.1: Aerial view of implementation site.

The crash-prone section runs parallel to I-35W, and short ramps allow transfers between freeways (Fig.3.1). The freeways and cross streets are labeled. The changeable message boards are denoted by the red circles and the high-volume on-ramp is enclosed by the green rectangle. The site includes two entrance and one major exit ramp and averages three lanes, with 3,000-foot auxiliary lanes in each of the two weaving areas. Weaving is excessive where high volumes enter from the ramp outlined by the green rectangle in Fig. 3.1, which combines traffic from I-35W, MN 65, and the downtown business center to the north. In addition to the traffic volume entering the rightmost lane from the ramp, many drivers on I-94 merge right in order to be in or near the rightmost lane in order to exit via two ramps downstream of Lasalle Ave. During periods of congestion, shockwaves generally originate when vehicles merge onto I-94 from the aforementioned ramp and cause vehicles in the rightmost lane to brake. If conditions are sufficiently dense, one vehicle entering from the ramp can initiate a chain reaction of vehicles braking that grows into a sharp shockwave that causes increasingly intense braking as it moves upstream. Due a vertical and horizontal curve, vehicles between Chicago Ave and Portland Ave have limited forward sight distance. Drivers' inability to see a shockwave approaching forces them to depend heavily on their reaction time to avoid rear-ending the vehicle ahead of them. Since driver distraction is one major cause factor of the crashes, detecting the crash prone conditions and warn the driver about them can increase their alertness and hopefully eliminate the driver distraction factor and keep some crashes from happening.

In an attempt to reduce the number of shockwaves, a double white line was extended by 1000 feet from the point where the ramp joins I-94 to the region between 1st Ave S and Nicollet Ave. The intent was to prohibit merging for another 1000 feet in order to move the merging zone to an area with longer forward sight distances and make it easier for drivers on the ramp to match the speed of traffic on I-94. This solution, although it reduced the severity of the crashes, it did not reduce their frequency.

Initially, for the purposes of projects related to intelligent transportation systems (ITS), a traffic detection and surveillance laboratory was established in 2002 as part of the Minnesota Traffic Observatory (MTO) of the University of Minnesota. Hourdakakis et al. (22) report details of the site instrumentation and capabilities. Because the site is close to the downtown area, nearby high-rise buildings allowed the installation of several cameras and machine vision sensors overlooking the roadway. While most of the sensors are not utilized by the queue-warning system, they provide the means of collecting detailed observations and determining ground truth.

As shown in Table 3.1, different types of data were used across this study. Individual vehicle measurements of speed and time headway were used for the operation and evaluation of the system. Aggregated traffic speed data from loop detectors serve for system adjustment as well in the system evaluation. In order to collect the necessary individual vehicle data cost-effectively and unobtrusively, machine vision detectors (MVDs) were utilized (Fig. 3.2). 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.

This study 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 (red circle in Fig. 3.1). Figure 3.3 shows the approximate coverage of the four cameras located on the roof top. Three of these four cameras are essential in the data collection of this work and their corresponding views are shown in Figure 3.4. The view of the 5h camera that monitors the sign is shown in Figure 3.5. 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 3.1: Summary of data types and purpose

<b>Data</b>	<b>Type</b>	<b>Source</b>	<b>Purpose</b>
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 methodology, algorithm and system design and development



Figure 3.2: Machine Vision Sensors

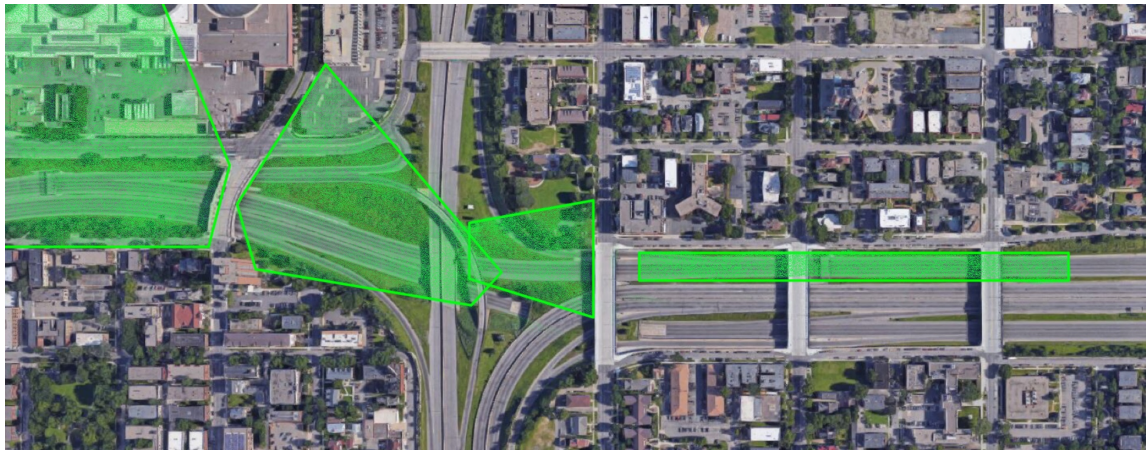


Figure 3.3: Coverage of the four cameras located on roof tops

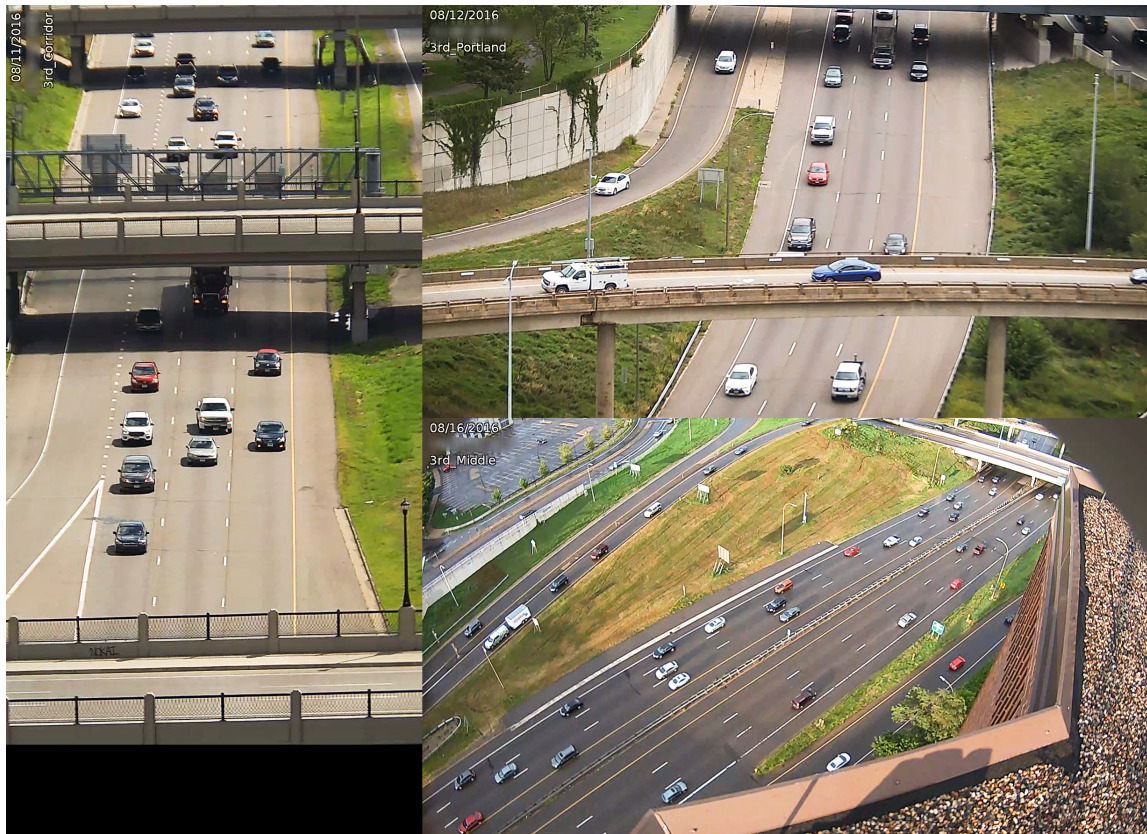


Figure 3.4: Views of the three essential cameras located on roof tops



Figure 3.5: The view of the sign-monitoring camera

## 3.2 Datasets for Off-line Methods

Static datasets are required in this study to develop different models and calibrate their parameters. Based on historical and real-time data mentioned above, two major datasets were created by this study, including I94QW2016 and MnFW2016. I94QW2016 includes the individual vehicle measurements and the record of conflict events. MnFW2016 covers the average speed data collected by loop detectors.

### 3.2.1 I94QW2016 Dataset

The I94WQ2016 dataset includes two major components: individual vehicle measurements and crash records of the study site described before. This dataset covers 55 workdays from June to August in the year of 2016, which were selected from the three month period according to their completeness, since data and videos are missing for some of the days due to hardware issues. There are 184 conflict events during this period with 41 crashes and 143 near crashes. The individual vehicle measurements were collected using machine vision sensors.

This work would like to acknowledge the hard work of many undergraduate students who watched through all the videos on this study site and recorded all conflict events manually, which contributed a lot in the collection of high-quality accurate conflict-event data. For each of the 55 days, there are 11 hours of videos from each of the four cameras(three essential traffic-monitoring cameras and one sign-monitoring camera). For the 55-day evaluation period, 2,420 hours of videos were manually watched in which the state of the sign and all conflict events were extracted. The author would also like to acknowledge the contribution in management and organization brought by some staff members at Minnesota Traffic Observatory(MTO).

### **3.2.2 MnFW2016 Dataset**

The MnFW2016 dataset was created based on the speed data collected by detectors along I-94 through the Iris system of MnDOT. Average speed of detectors along the study site of 30s were extracted to formulate this dataset. The MnFW2016 dataset were used in determine the external control and system adjustment.

## **3.3 Data Preprocessing**

### **3.3.1 Observed and Potential Data Issues**

Off-line, static datasets can be very different from real-time data, even if they are collected by the same set of detectors. There are a number of issues regarding the raw data in both I94QW2016 and MnFW2016. The main issue with MnFW2016 is that if no vehicle was detected in a 30s interval, the detector returns -1 as the speed. Such an issue is easy to deal with. One of the processing strategy is to estimate the speed with the free-flow speed such as using the speed limit. The issues on the individual vehicle measurements are more complicated and hard to deal with. This subsection addressing this issues and provides an analysis of the reasons of such issues. Some of the issues are related to the design of the machine vision sensors and their communication algorithms. But these issues can also exists in other sensor that provides individual vehicle measurements in real-time. Since the datasets were collected by the real-time system, the real-time system can face some of the same issues. However, thanks to the robust design and the capability of controlling the sensors in real-time, the on-line system can deal with most of the issues while off-line systems cannot.

### **3.3.2 Duplicated Data**

One problem of the data collected by the Machine vision detectors is that they contain pieces of duplicate data. These duplicate data entries are identical to data collected before with the same time stamp. These data usually exists in batches instead of single entries.



The reason of this problem is due to communication disruptions. The machine vision sensor used in this study was AutoScope. Each detector out on the study site has a buffer to store collected data. Ordinarily, only new detected data will be delivered to the control center miles away from the study site. However, when the communication between the control center and the sensors are disrupted, the sensors get confused about which data has already been delivered to the control center. Therefore when the communication is restored, all the data in the buffer are dumped to the control center, resulting in duplicated data blocks. For the real-time implementation, the system developed in this study has control of the sensors and can stop the dumping of old data as soon as it is detected. The system also stops using the corrupted data when detected. However, since an off-line model does not have such capability, it needs preprocessing to remove the duplicated data to solve this issue. The reason that the raw data were collected and recorded for off-line uses is that it is important to collect raw data which fits for identifying all sorts of problems for future implementation and improvement.

### 3.3.3 Missing Data

Individual vehicle measurements is in a form of a data entry for each vehicle detected. No trace will be left when a vehicle entry is missing. One possible method to detect missing data entries is to examine the dataset with the surveillance footage of the study site, which is not feasible for “massive production”. There are a number of reasons for missing data. One is because the sensor didn’t detect a vehicle, another is the data was lost during the communication. The author found that if a data entry was lost due to communication issues, it is possible to detect some of the missing data and recover them. The data were collected by the sensors, this means the lost data entry can be still in the buffer on the study site. If the communication issues trigger a data dump (the process that the sensors dumps duplicated data to the control center), the missing data entry can be dumped to the control center. In this case, a unique data entry can exist in the data dump, which should not be removed. Instead, such a data entry should be placed between data entries collected before and after it according to its time stamp.

### 3.3.4 Temporal Alignment

Another problem is temporal alignment between the two stations. Since the methodology utilizes two detector stations and each station maintains its own clock, the time between the two stations may not be perfectly aligned. Several difficulties can be brought by the imperfect alignment of the data. In order to quantify the traffic condition, the data used from both detector stations should not be very far from each other, otherwise the measurements may not be accurate. Such inaccuracy can cause huge problems in the recognition of crash-prone conditions and result in wrong results or illusive patterns. For an on-line system, such a problem is not a big deal since the individual vehicle measurements are fed into the system as soon as they are collected. The system can simply “trust” the clock of one detectors. For a off-line system, this may not be the case. One cannot even sort the data based on their clock since their clock are not the same. Also, if the sensor station has the ability to perform adjustment of time through the Internet, it may introduce variants drift between the detectors since the accurate of Internet time synchronization is not perfect and can introduce different errors for both stations.

Two different approaches can be used to resolve the problem: to improve the sensor to get better data and cope with current data. Improvement of the sensors usually need to happen before data collection because once one gets the data, this method cannot help with the collected data. To improve the sensor, accurate time adjustment methods such as using GPS modules to get accurate time can be applied on hardware level to avoid the problem. On software level, one can create a time-stamp when the data are received or collected according to the data collection server. However, sometimes these may not be viable options since the data are already collected by the sensors. For the data used in this study, the clock of the two detector stations has Internet time synchronization multiple times every day. Therefore, the difference between the two clocks changes when either of the stations performed such time sync. Fortunately, millisecond ticks were kept with the data to provide the time difference in high resolution. The update of such ticks is usually 1 millisecond, but each device can have individual differences in the frequency of the process or time keeping unit. This causes a difference in tick counts even if two devices started counting from zero at the exactly same time. One problem with such ticks is that it will

reset whenever there's a power loss or system reboot. To adjust the times with these issues, this study adjusts the times based on the ticks and estimates the reset ticks according to the clock.



## Chapter 4

# Traffic Measurements and Metrics

In order to identify CPCs, it is crucial to have a proper representation of traffic conditions. As individual vehicle measurements hold the benefit of having 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 or totals. 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 in this study, 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. [4] 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 study 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 study, 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.

## 4.1 Temporal Metrics

### 4.1.1 Average Speed

Average speed is a common and informative statistic and helps reduce stochastic noise. It shows the average level of speeds during a given time interval. The calculation is shown in Equation 4.1.

$$AvgSp = \frac{1}{n} \sum_{i=0}^n u_i \quad (4.1)$$

Where:

$n$ : number of vehicles

$u_i$ : speed of vehicle  $i$

### 4.1.2 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 as shown in Equation 4.2. 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.

$$CVS = \frac{\sigma}{\mu} \quad (4.2)$$

Where:

$\sigma$ : standard deviation of vehicle speeds

$\mu$ : mean of the vehicle speeds

### 4.1.3 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. Equation 4.3 shows the calculation of time headways and Equation 4.4 shows the calculation of the coefficient of variation of time headway.

$$h_i = t_{i+1} - t_i \quad (4.3)$$

Where:

$h_i$ : time headway between vehicle  $i$  and vehicle  $i + 1$  that detected after vehicle  $i$ .

$t_i$ : the detection time of vehicle  $i$

$$CVTH = \frac{\sigma}{\mu} = \frac{\sqrt{\frac{\sum_{i=0}^n (h_i - \bar{h})^2}{n-1}}}{\bar{h}} \quad (4.4)$$

Where:

$\sigma$ : standard deviation of time headways

$\mu$ : mean of the time headways

$\bar{h}$ : mean of the headways

$h_i$ : time headway between vehicle  $i$  and vehicle  $i + 1$  that detected after vehicle  $i$ .

## 4.2 Spatial Metrics

For the purposes of this study, spatial metrics refers to measurements derived from the trajectory information of a single vehicle over a specified section of roadway. While such measurements are hard to collect, they can be very descriptive of the state of the flow. These metrics can be directly measured by wide area sensors like radar when available. This study utilized a single detector station approach to estimate these metrics from individual vehicle measurements. The estimation was based on the following assumptions [29]:

- The space headway between vehicles remains fixed after they depart from the detector. Given that and the length of each vehicle, the number of vehicles that fit in a hypothetical one-mile of road can be calculated.
- Vehicles keep constant speed while inside this hypothetical one-mile segment same as their speed over the detector unless,
- The preceding vehicle is moving with a lower speed in which case the subject vehicle will match speed while inside the segment.

Each time the sensor collects a point measurement of a vehicle, that information and the information of all preceding vehicles within the hypothetical one-mile segment are used for the estimation of the spatial metrics.

### 4.2.1 Density

Density ( $k$ ) is defined as the number of vehicles per unit length of a road segment. 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 study, density is not used directly but rather as a component in the calculations of other traffic metrics such as traffic pressure and kinetic energy.

Density is estimated through Equation 4.5, where TTT stands for total travel time defined by Equation 4.6. By combing both equations, the density is computed through Equation 4.7.

$$K = \frac{\sum t_i}{t \times dx} = \frac{TTT}{t \times \Delta x} \quad (4.5)$$

Where:  $K$ : density

$t_i$ : travel time of vehicle  $i$

$t$ : time

$$TTT = V \times \bar{T}_t = \frac{V \times \Delta x}{\bar{u}_t} \quad (4.6)$$

Where:

$TTT$ : Total Travel Time in time  $t$

$V$ : Average volume in time  $t$

$T_t$ : Average travel time

$u_t$ : Time mean speed

$$K = \frac{\sum t_i}{t \times dx} \quad (4.7)$$

Where the notation is the same as before.

#### 4.2.2 Acceleration Noise

Acceleration noise is a measure of the smoothness of the traffic flow based on an estimation of individual acceleration dispersion. Three factors are highly related to the value of acceleration noise: driver, road and traffic condition. The calculation of the acceleration

noise in this study follows the approximation proposed by Jones and Potts [23] as described in Equation 4.8.

$$A_N \approx \sqrt{\frac{1}{T} \sum_{i=0}^T \left(\frac{\Delta u}{\Delta t}\right)^2 \Delta t - \left(\frac{u_T - u_o}{T}\right)^2} \quad (4.8)$$

Where:

$u_i$ : Speed of vehicle  $i$

$u_o, u_t$ : Speed of the first and last vehicle in the hypothetical one-mile segment

$T = t_T - t_o$ , where  $t_o, t_T$  arrival times of first and last vehicles at the measuring point

### 4.2.3 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 [24] proposed another measurement, the mean velocity gradient, described by Equation 4.9.

$$MVG = \frac{\sum_{i=1}^N (MVG_i)}{N} \quad (4.9)$$

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

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

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

### 4.2.4 Quality of Flow Index

The quality of flow index proposed by Greenshields [25], 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 4.10).

$$QFI = \frac{\sum_{i=1}^N QFI_i}{N} \quad (4.10)$$

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

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

$\bar{u}$ : Average speed

$\Delta u$ : 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.

#### 4.2.5 Traffic Pressure(TP)

Traffic pressure (TP) was designed to measure the smoothness of traffic flow. It is defined as the product of speed variance and density [26] as seen in Equation 4.11. 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.

$$TP = \sigma_s^2 \times k \quad (4.11)$$

Where:

$TP$ : Traffic Pressure

$\sigma_s^2$ : Speed variance

$k$ : Density

### 4.2.6 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 [27], 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 4.12.

$$KE = ak(\bar{u})^2 \quad (4.12)$$

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

## 4.3 Heuristic Metrics

### 4.3.1 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.

### 4.3.2 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.

### 4.3.3 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.

### 4.3.4 Individual Vehicle Speed Noise Reduction

All sensors inherently have error and noise in their measurements, and such noise can cause surges in the crash-likelihood calculation and compromise the accuracy of the model. In order to reduce the noise in a time series, filtering techniques such as the one proposed by Hourdos [28] in his development of a crash probability model are often employed. To perform a time series analysis and noise removal, the original unstructured speed data needs to be translated into a 1-second-speed time series. To achieve this goal, two different interpolation methods, linear interpolation and spline interpolation, were considered as candidates and tested. As shown in Figure 4.1, linear interpolation assumes the data between individual vehicle speeds follows a linear relationship. Figure 4.2 shows the spline interpolation connects the points with a smooth curve. When comparing these two methods, spline interpolation was found to be problematic as the requirement that the data

points be connected with a smooth curve produced interpolated speeds that were well outside of the normal range thereby introducing additional error and noise. As in the example presented in Figure 4.2, spline interpolation can introduce extreme values and in some cases it can generate huge spikes that are outside the feasible region of vehicle speeds. Because of such an issue of the spline interpolation, the linear interpolation was selected for this study. Of the several different filters that Hourdos [28] designed and tested on their ability to remove impulse noise, his Digital FIR Hamming filter was selected to perform the noise reduction for the system.

After filtering, the data become another time series with lower noise. Given that the time headways between vehicles are as informative as their speeds, the dataset needs to be returned to its original form before the traffic metrics are calculated. A reverse interpolation method finding the filtered speeds at the times of the original data points is implemented.

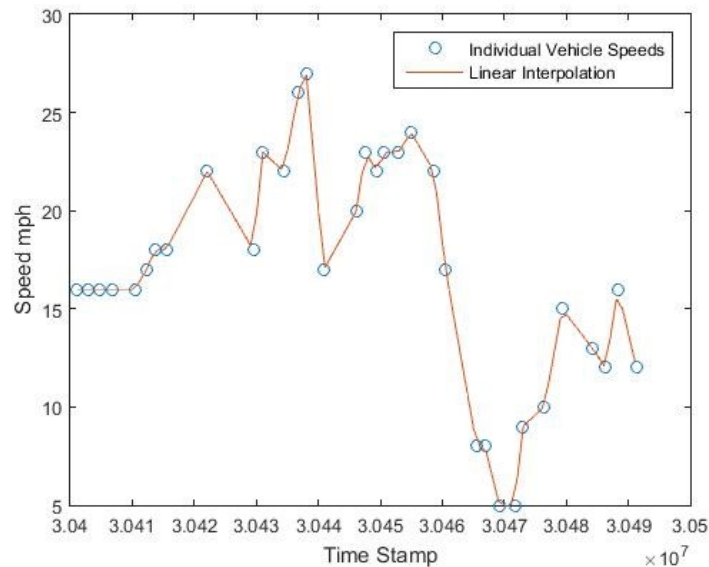


Figure 4.1: Linear Interpolation

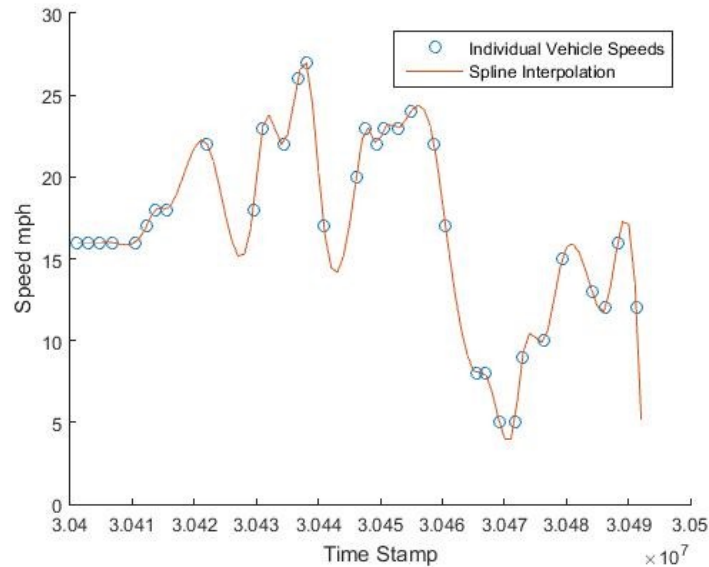


Figure 4.2: Spline Interpolation

#### 4.4 Extension in Space and Time

Traffic shockwaves can be triggered by several different things and once they have been triggered, it may take time for traffic to deteriorate enough to be considered crash prone. Therefore, the initial triggers of events that lead to rear-end crashes can be far away from the actual crashes, both in time and space. Simply using the most recent traffic metrics may not be enough to identify the pattern of CPCs. By also utilizing metrics at various past intervals, the signature patterns of CPCs can begin to be recognized. For example, when a crash occurs, point measurements can provide the average speed and acceleration noise just before the crash. However, since the crash can be the product of a sharp shockwave generated downstream of the crash location two minutes prior, just using the traffic conditions at the time of the crash at the exact location of the crash may not be sufficient to reveal the pattern of conditions that lead to it. Towards that effect, the methodology utilizes a multiple moving windows approach. Window size determines how many vehicles need to be considered in the calculation of each metric while the location of the window in the data stream is determined by a prior time shift from the present

time. Specifically, the number of vehicles included in each window is selected from the set 15,30,40,50,60,70,80,100,110,120 for use with the temporal and spatial metrics. To capture the traffic conditions of different groups of vehicles in the past, the size of the time shift in seconds is selected from the set {10,30,60,120,180,240,300}. Given a pair of the two parameters, a window containing several vehicles in the past (or at present) is determined. The traffic metrics introduced in the last section can be calculated to represent the traffic conditions for each group of vehicles. Using all of the different metrics with their variants, a 1-by-1,930 vector is created to represent a given traffic condition in great detail. This representation was used in this study to investigate the pattern of CPCs.

# Chapter 5

## System Design

### 5.1 Multi-layer Design

As shown in Fig. 5.1, the system follows a three-layer design. The Dangerous Level Measurement Layer(DLML) handles the collection of raw data and then using them to estimate the dangerous level of the real-time traffic conditions. The noise reduction of individual vehicle measurements happens in this layer. Communication servers are in charge of receiving data from different detector stations and unify them under the same structure. The polling client is a program agent request merged data from communication server and pass it to the computational servers to handle all the calculation, including the filtering, calculation of the real-time traffic metrics and the estimation of the level of dangerous. the the measurement of the dangerousness used in this study is a crash probability model proposed by Hourdos [28].Therefore, a crash probability will be calculated for each real-time traffic condition, measuring the dangerous level of it. This crash likelihood along with additional traffic information such as speeds and headways are passed to the second layer, the Queue Warning Algorithm Layer(QWAL).

In the QWAL, an algorithm is embedded for determining the result of the warning messages based on the dangerous level measurement and some other information about the traffic flow delivered by the DLML. The decision is made through a reasoning model in the algorithm that can determine a suggestion of the alarm state based on the information

collected.

Such suggestion then gets passed through the system oversight layer(SOL) where the suggestion of the alarm state is judged with a set of rules carried out by the policy makers. Then a final result of the warning messages will be delivered to the active message sign on the highways. The final alarm result can be different from the suggestion of the SOL according to the rules proposed by the policy makers.

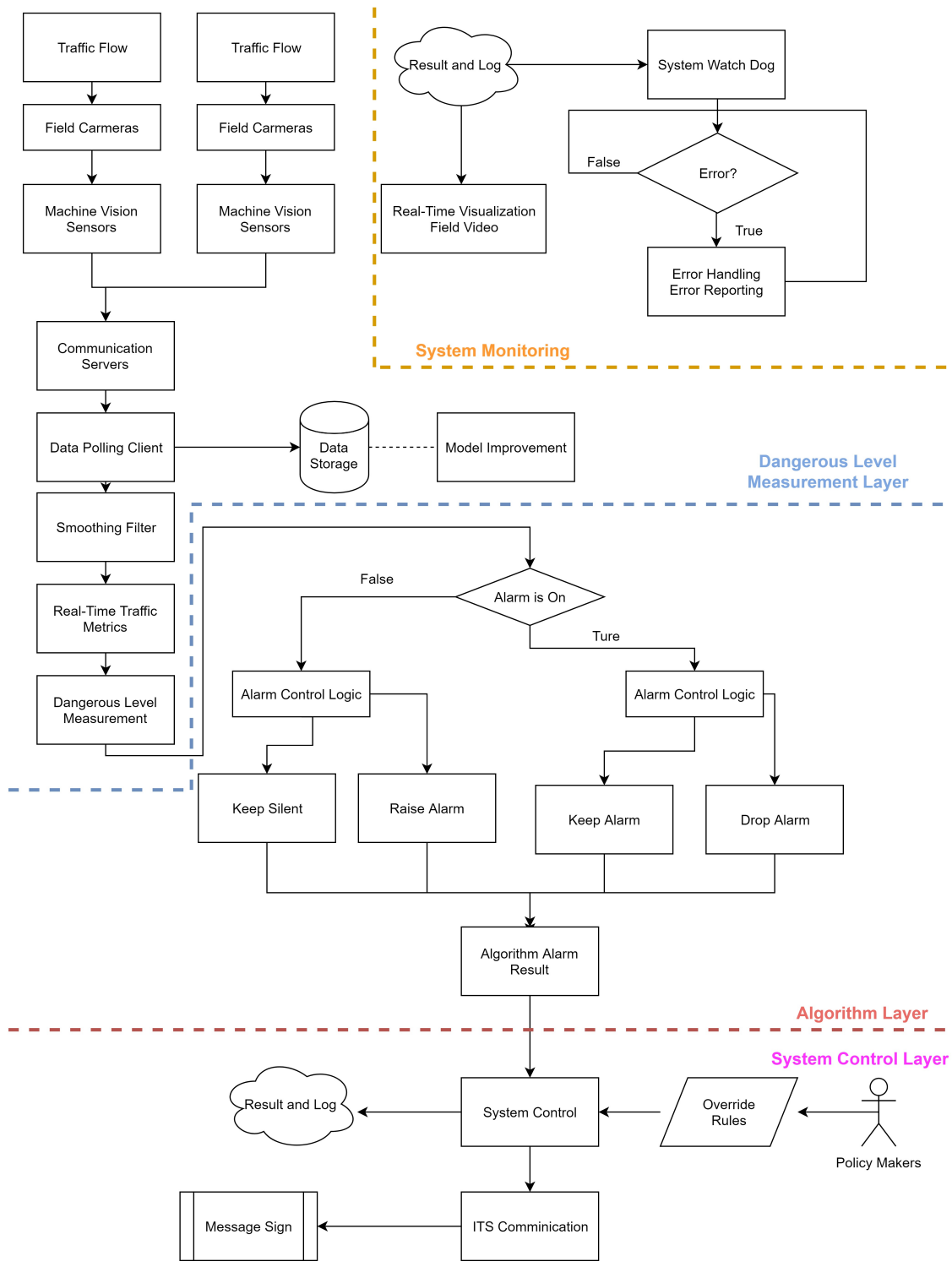


Figure 5.1: System Architecture

## 5.2 State Machine Formulation

When a system works in real-time, it is crucial for it to have the ability to handle abnormal conditions and react accordingly. In the operation of the real-time queue warning system, a lot of unexpected conditions can occur that can have a huge impact on the different components of the proposed system. For example, disturbances on the radio and Ethernet communication can lead to non-responding detector stations. Malfunction of detectors such as power lose, blocked vision etc, will also result in missing or incomplete data. During such “communication blackout” of certain number of real-time data sources, the result of the system under normal working condition may no longer be accurate due to missing or incomplete data. The system should be aware of such abnormal situations and perform proper reactions such as stop providing warning messages and notify the administrators about the situation.

As a real-time system, the queue warning system proposed should be robust and able to handle most of the abnormal situations. In order to achieve this, a state machine formulation was applied in the work-flow of the proposed queue warning system. As in Figure 5.2, four different states were designed for the system to handle both normal and abnormal working conditions. State 0 can be considered as the preparation state. State 0 represents the “booting” processing of the system. When the system is inside this state, it means that there are not enough data to generate the necessary metrics. For example, some of the traffic metrics may requires 120 vehicles 300s before and these amount of data won’t be available immediately after the start of the entire system. In this case, the system will wait and just collect data until there are sufficient amount of data to perform the analysis.

When sufficient amount of data are collected, the system will exit State 0 and enter State 1, Working State. When the system is inside this state, the process presented in the previous chapters will proceed. The tasks including individual vehicle measurement noise reduction and all others from all the different layers will be operating normally. And only in State 1, active messages that warns the drivers about dangerous conditions will be delivered to the sign. If the system is not in State 1, no alarms will be delivered to the sign.



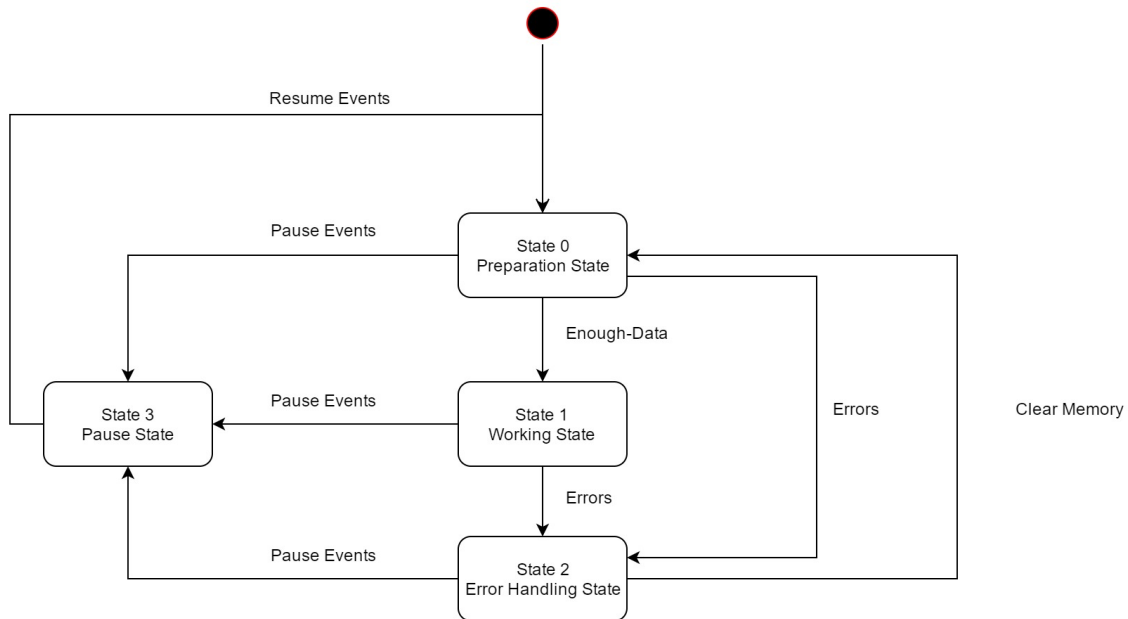


Figure 5.2: State diagram of the system

When errors detected in State 1, the system will exit State 1 and enter State 2, the error handling state. There can be many sources of errors one example of which is the “communication blackout” discussed above. Others include errors in mathematical calculation, dis-synchronize or old data reporting, etc. When the system is in State 2, it will perform actions such as rebooting the detectors and clear the memories trying to restore the system back to normal conditions.

While the system is in any of the State 0, State 1 and State 2, a pausing event can make the system exit its current state and enter State 3, the pausing state. In State 3, the only process that is going on in the system is the data collection. Other process remain in hibernation until a wake up command is issued. Having an pause state can be very helpful to keep the data collection process going but pausing the working flow. A pausing event can be the system failed to establish connection with the detectors after a large number of trails. An application of State 3 in the proposed system implemented on I-94 is to put the system in State 3 during midnight and wake it up in the morning, since the traffic conditions at mid night are usually not CPCs.

With the state machine formulation, the system can handle abnormal situations and avoid providing alarm messages based on missing or incomplete data. It is essential in the operation of the proposed queue warning system.

### 5.3 Dangerous Level Estimation

To determine if a given traffic condition is crash prone or not, a measurement of the dangerous level of traffic conditions is desired. Many different approaches can be used to estimate the dangerous level of traffic conditions and one of these approaches that often applied is statistical regressions that model the probability of a crash. The higher the crash probability, the more dangerous the given traffic condition is. Therefore, a dangerous level function  $Da(Tr)$  is desired the value of which would increase with the increase of the dangerous level of given traffic condition.  $Tr$  represent a given traffic condition, in this case it is a vector of values of all/selected metrics proposed in Chapter 4.

### 5.4 Queue Warning Algorithm

This section describes the second layer of the system, the Queue Warning Algorithm Layer(QWAL or QWA Layer). The QWA is designed to generate a suggestion of the warning messages. In other words, it decides whether an alarm should be generated, when the alarm should be generated, how long the alarm should last once generated and what message need to be delivered to the message sign. The algorithm takes the result of the DLML and necessary traffic measurements, then base on theses information and its reasoning logic, it will determine if a warning result should be generated. The flow chart of the proposed algorithm can be found in Figure 5.3. Another moving median filter, or average filter, is applied to the dangerous level measurement to reduce noise and outliers before it get to used in the QWAL.

A dynamic window approach is proposed to increase system's tolerance on spikes in the dangerous level measurements. This thesis will show the importance of this approach

and the impact from spikes on the system later in this section. An adjusted dangerous level measurements is calculated for real-time traffic conditions using the dynamic window approach. Based on this adjusted value, user-defined thresholds, and the result of a speed test, the suggestion 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 minimum time extension.

The dangerous level measurements alone may not be sufficient for determining the precise time to warn the drivers while still producing consistent and stable decisions. Since the such measurements are usually continuous variables, a two-threshold approach is employed to determine whether a condition is dangerous enough that worth the warning. One threshold for determining the raising of the alarm and a second for determining its termination. In addition to the dangerous level measurements, the algorithm proposed in this study 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, this study calls this Minimum Alarm Length. 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 spike here means one single value that is extremely large with very low values bot behind and in front of it. If the dangerous level measurements are sufficiently low both before and after the spike, it implies this spike may be an outlier and the condition is not dangerous enough for warning. Spikes in the measurement of dangerous level can be hard to handle for a real-time queue warning system. This is because that a spike can only be determined as a spike after it occurs. In historical data, spikes are easy to identify since the information after the extreme point will also be available. However, in a real-time system, when an extreme point is reported, it can be hard to determine if such extreme point will be a spike or a CPC has been created within the detection zone. If such an extreme value is a spike, raising the alarm may cause

a false alarm lasted for the minimum alarm length. On the other hand, if such extreme value is a dangerous condition, ignoring it can result in failure to warn the drivers in time. The dynamic window approach was proposed to solve this issue for the proposed real-time queue warning system.

The key idea behind the dynamic window approach is to determine the level of filtering based on the raw value of the latest result. Inherently, the more data points in the average filter window, the more historical information will get included in the filtered result. It means, the latest data points will have a lighter weight in the final result. Therefore, although with bigger average window, the impact of noise will get reduced, the delay in the filtered value will increase. The dynamic window approach changes the size of the averaging window based on the latest dangerous level measurements. In the implemented system on I-94, the window size will become smaller if the latest measurement is higher. With this approach, the system gained the ability to be more responsive when the latest dangerous level is high. The control logic of the algorithm can be found below in Equation 5.1.

$$\begin{aligned}
 & Greater(\bar{p}_t, \lambda_1) \wedge SpeedTest(v_{tb}) \rightarrow AlarmOn \\
 & AlarmOn \wedge Greater(\lambda_2, \bar{p}_t) \wedge TimeCheck(t) \rightarrow AlarmOff \\
 & SpeedTest(v_{tb}) = \begin{cases} true & v_{tb} \leq u_1 \\ false & v_{tb} > u_1 \end{cases} \\
 & TimeCheck(t) = \begin{cases} true & t - t_0 > \Delta t \\ false & t - t_0 \leq \Delta t \end{cases}
 \end{aligned} \tag{5.1}$$

Where:

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

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

$\lambda_1$ : Starting Threshold

$\lambda_2$ : Ending Threshold

$v_{tb}$ : Current Downstream Speed (mph)

$\bar{p}_t$ : Adjusted Crash Probability

$$\bar{p}_t = \frac{1}{n} \sum_{i=t-n+1}^t p_i$$

$$n = \begin{cases} f(p_t, \lambda_1) & p_t > \lambda_1 \\ g(p_t, \lambda_1, \lambda_2) & \lambda_2 \leq p_t \leq \lambda_1 \\ h(p_t) & p_t < \lambda_2 \end{cases} \quad (5.2)$$

As mentioned above, the algorithm regards the dangerous level measurement as one of the inputs needed to decide whether to raise or lower the alarm. The adjusted dangerous level measurement is used instead of the direct output from the crash probability model. The value  $n$  is the size of the dynamic window used to average the crash probability. Using a dynamic window size makes the algorithm more sensitive to crash-prone conditions and reduces the probability of false alarms caused by noise in the measurement. The dynamic window size is calculated through three conditions, as described in Equation 5.2, with different results depending on which threshold it is nearest. The controlling hypothesis is that the noise in the crash probability will be higher around the selected thresholds so more samples are used to calculate the average unless the crash probability suddenly increase to values close to 100% in which case the averaging window is much smaller in order to reduce the delay in raising the alarm.

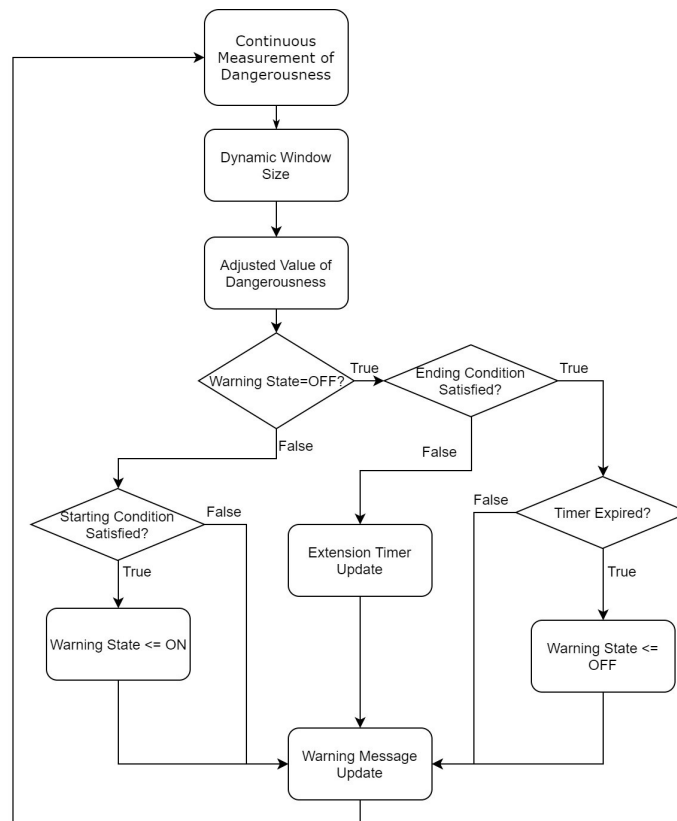


Figure 5.3: Flowchart of the Queue Warning Algorithm

## 5.5 System Control

As described before, the queue warning algorithm will provide a suggestion of whether to raise the alarm or not. Such suggestions are not the ultimate result being delivered to drivers. As an Active Traffic Management(ATM) system, policy makers may desire certain rules or controls on the final result. Also, visualization tools are desired for better monitoring of the system. The last layer of the proposed system provide the room for certain policy controls and monitoring the system.

### 5.5.1 Policy Adoption: Override Rules

To provide the control of the alarm result to policy makers, the system has a list of override rules that will make the system have different final results compared to the alarm suggestion provided by the queue warning algorithm. Based on the information required by the override rules, they can be classified into three categories: 1) Time-based Overrides, 2) Condition-based Overrides and 3) Complex Override functions.

Time-based override rules are rules that change the decision of the algorithm at certain times. The time can be specified by time of day, certain dates, whether it is a week day or not and so on. The intuition of such time override is to add preference to the system to make it silent at certain time event if the traffic conditions are dangerous. Condition-based override rules are to keep the system silent under certain traffic conditions even if the traffic conditions are dangerous. One example can be the sign are too far from the detection zone and the policy maker prefer the sign not raised when the traffic under it is congested. Complex override functions can be considered as a general form of override rules which provide an override decision based on time, traffic condition and the state of the alarm. Such function provide policy makers the capability of embedding certain preferences and total control of the output of the proposed system. This general form is described as Equation 5.3. Imagine an extreme case that policy makers want to invert the alarm, they can simply set a function such as  $f(S(\mathbf{Tr})) = 1 - S(\mathbf{Tr})$  to be the override rule.

$$Alarm(\mathbf{Tr}, t) = f(\mathbf{Tr}, t, S(\mathbf{Tr})) \quad (5.3)$$

where:

$\mathbf{Tr}$ : Traffic condition representation

$t$ : time

$S(Tr)$ : The suggestion provide by the queue warning algorithm; 1 means to raise the alarm, 0 means to drop the alarm.

$f(Tr, t, S(Tr))$ : An override function with similar output of  $S(Tr)$ .

$Alarm(Tr)$ : Final alarm result.

### 5.5.2 System Monitoring

For the ease of monitoring the queue warning system, visualization tools and interfaces were designed for the proposed queue warning system. Several key components are proposed by this study to provide administrators real-time information of the system. These interfaces include real-time data pool, active system logs, visualization of traffic dangerous measurement and alarm decisions, real-time footage of the traffic conditions. Readers can find more details about an implementation of the system monitoring tools in Chapter 6.



## Chapter 6

# System Implementation

### 6.1 Dangerous Level Estimation: A Crash Probability Model

In this study, a crash probability model proposed by Hourdos [4] was tested as the dangerous level measurement in the proposed real-time queue warning system. The parameters and coefficients of the model can be found in Table 6.1. The system collects individual vehicle measurements and calculates the corresponding traffic metrics proposed in Chapter 4. These traffic measurements are converted into a crash probability using this statistical model to measure the level of dangerousness of the real-time traffic conditions.

Table 6.1: Crash Probability Model [4]

Variable	Estimate	Std. Error	p-Value
<i>Constant</i>	28.4417	17.116	0.0962
$AS - 120 - 300 - d$	0.22867	0.06191	0.00020
$AS - 15 - 120 - u$	0.32461	0.12515	0.00950
$(AS - 15 - 120 - u)^2$	0.00704	0.00262	0.00710
$AS - 15 - 30 - d$	0.27744	0.10227	0.00670
$(ASM - 30 - 10 - u)^2$	0.00421	0.00091	0.00000
$CV - 110 - 10 - d$	49.1112	21.9112	0.02500
$KE - 15 - 10 - d$	0.00008	0.00004	0.02790
$\log [CV - 110 - 10 - d]$	12.5625	5.58787	0.02460
$\log [KE - 15 - 30 - d]$	4.62125	1.31443	0.00040
$MaxMinDiff - 30c - d$	0.05188	0.01637	0.00150
$MVG - 120 - 300 - u$	134.746	40.6807	0.00090
$TP - 100 - 30 - u$	0.00195	0.00098	0.04740
$TP - 110 - 10 - u$	0.00260	0.00102	0.01060
<i>Pavement</i>	0.80218	1.08341	0.45900
<i>Sunnight</i>	1.89274	1.15225	0.1005
<i>Sunside</i>	0.658823	0.7118	0.3547
<i>Sunfront</i>	1.19134	0.5953	0.0454
<i>Visibility</i>	0.39735	1.18592	0.73760

**Notation:** metric-window size-time offset-station; u = upstream and d = downstream; AS = average speed right lane; ASM = average speed middle lane; CV = coefficient of variation of speed; KE = kinetic energy; MVG = mean velocity gradient; TP = traffic pressure; pavement = dry (0) or wet (1); visibility = clear (0) or reduced (1); sun position = night, cloudy (aliased), sun in back or side, sun in front. A detailed description of the metrics and the model can be found in a previous study by the author [4].

## 6.2 Details of the Implemented Prototype

This study deployed the proposed system on the right lane on the I-94 westbound freeway segment at the study site. The values of many parameters need to be specified based on the study site and may need to be optimized for different implemented location to get the best warning performance. Observations on this study site shows the travel time of shock-waves from the detection zone to the variable message sign at upstream is usually about one minute. Therefore the minimum alarm length was set to be one minute. For the on-line system, one set of thresholds was selected (0.035,0.035) for testing. Different sets of thresholds were tested in the off-line version of the system to show the performance. The dynamic window size in this prototype was a simple linear model determined by  $\{(0.5\lambda_1, 20), (1, 5)\}$ . The implemented prototype was running on a two-second interval. This means the system will collect data from detectors and perform all the operation every two seconds.

This prototype utilizes the variable message signs deployed by MnDOT. The signs were integrated in their general control system which runs in a thirty-second update interval. This means the sign get updated every thirty second. If a new message arrives right after an update, it has to wait thirty seconds to be delivered to the sign. During the testing of the implemented prototype, since there are other factors such as communication delay, it is observed that the delay between the system and the sign can be 30 seconds to 1 minute. In ideal implementation situation, such delay can be reduced by proper design of the sign update protocol that fits better for the proposed system. I this implemented prototype, it was not feasible to modify the internal communication system with the sign. In latter sections, this thesis will show the impact of such delay the impact of external controls on the system performance.

As presented in Figure 5.2 before, the prototype operates in corresponding states given different situations. There are several conditions that can trigger the system to go into state 2. If non of the detectors returns any data entry for 120 seconds, the system assumes there's some issues with the communication or the sensors and enters the error handling state. The implemented system ensures the millisecond ticks for all the detectors are reset daily and won't overflow in a twenty-four-hour period. Therefore if an decrease in the millisecond ticks is detected, the system assumes there are some issues with the detector(entered data dumping mode due to communication interference or some other error or bugs) and will enter state 2. Also, any exceptions such as arithmetic error will also trigger the error handling reaction of the system.

Besides the main alarm generation program, a watchdog utility was implemented to monitor the status of the main program. The watchdog will reset the main program every day at midnight when the prototype is in pausing state to avoid some potential long-run issues and let the main program re-initialize itself every day. It has the ability to detect problem such as program freeze, close or crash of the main program and restart the main program and notify the administrators accordingly. The watchdog utility was implemented using Python and runs parallel with the main program. It even increase the already robustness of the main program and its stability.

## 6.3 System Oversight And Policy Adoption

### 6.3.1 External Controls

As part of the terms for the deployment of the system, two overrides were included and the preexisting MnDOT signs' (Fig. 6.1) refresh rate was kept. Two sets of changeable message boards used are fixed to gantries above the freeway. The overrides are intended to limit possible overexposure of drivers to the warning by what was, at the time of implementation, an unproven system and consist of a time override and a congestion override. The time override prevents the sign from being turned on, regardless of the alarm status before noon or after 8 p.m. MnDOT felt that traffic conditions at those times would be too prone to producing false alarms. The congestion override also prevents the sign from being turned on when five consecutive 30-second average speed measurements at the loop detector before the farthest upstream sign are below the MnDOT-imposed threshold of 25 mph. This override is intended to reduce driver overexposure to the warning by not displaying a warning when drivers are already traveling slowly. The rate at which the sign is updated is a result of the sign being part of the MnDOT Twin Cities metro-wide network. Initial activation can vary from a few seconds to one minute, depending on the synchronization between the independent queue-warning system and the traffic operations system that controls the signs. This delay amplifies short gaps in the alarm activation (e.g. even if the alarm is only lowered for 10 seconds, the sign could be off for up to a minute).

### 6.3.2 Interface

In order to allow the users of the system to monitor the system and traffic conditions in real-time, a live feed of the model result, MVD data, sign status, and cameras used for event detection is available at any time that the system is operational. A continuously updating plot of the model result with the threshold for reference is also displayed with the data feeds. The message sign and the back-end system interfaces can be found in Figure 6.1, Figure 6.2 and Figure 6.3. The software is developed using C++ and Python. The plotting utility in the implemented system is powered by MATLAB. The system is designed

to run on Windows platform and has been tested on Windows 7 and Windows 10 servers. In order to assure the real-time performance, dual-core or multi-core CPU with a single core frequency higher than 3.0 GHz is recommended. Sufficient network connection and bandwidth is also required to avoid communication failures. A simplified Class Diagram is in Figure 6.4 that shows the basic implementation of the system.



Figure 6.1: MnDOT changeable message boards with warning displayed.

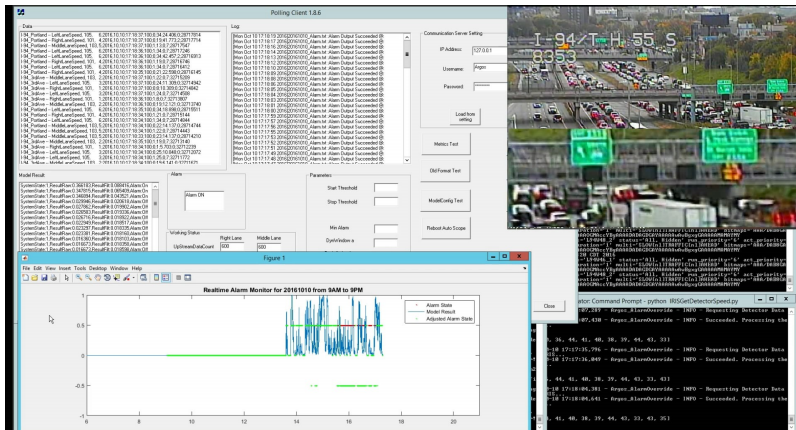


Figure 6.2: Back-end System Interface



Figure 6.3: Back-end System Interface

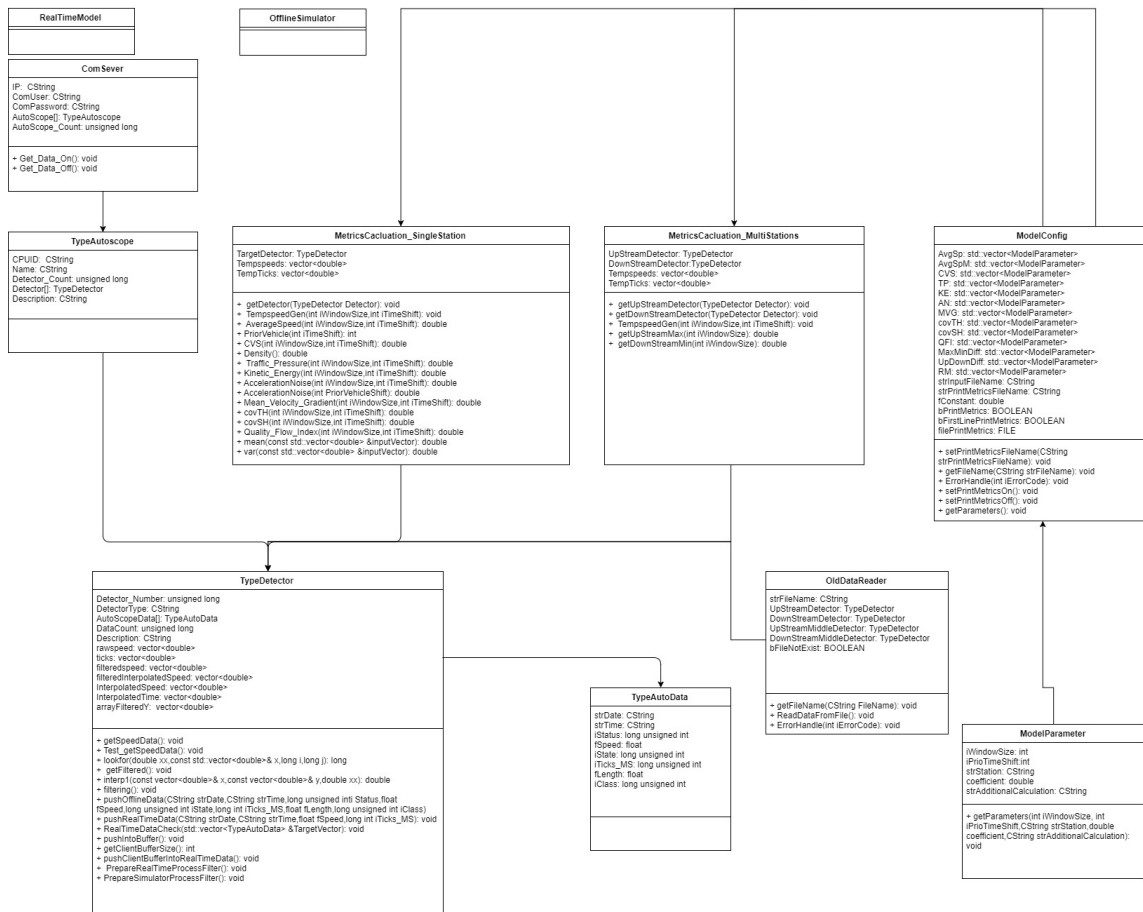


Figure 6.4: System Simplified Class Diagram

## Chapter 7

# System Evaluation

### 7.1 Traditional Evaluating Method

To evaluate a given warning system, the results generated by such a system are evaluated by the number of conflict events it detects. This is called the detection rate of the warning result as showed in Equation 7.1. This value shows the percentage of the conflict events successfully detected by the system. The higher this value is, the better performance of the warning result. However, solely rely on the detection rate can be puzzling. Considering a scenario that a system generates warning messages throughout the whole day. It would be able to detect every single conflict event since the alarm is always on. However, there are very long time during the time alarm is on but no conflict events can be observed. As an active warning system, such result will make it the same as a fixed warning sign. Such “active” warning messages can be discredited by the drivers and lost its credibility. To measure the level of non-conflict events inside the period the alarm is on, an index call false alarm rate was used in the literature. A false alarm is an alarm with no conflict event happened within. Equation 7.2 shows the calculation of false alarm rate.

If two systems have the same detection rate, the one with a lower false alarm rate is desired. Usually, there is a trade off between detection rate and false alarm rate since when a system is more sensitive to crashes, it can raise the alarm more easily. Therefore, a sensitive system tends to generate more and longer alarms, which can lead to a high false alarm



rate. This, however, can vary for different scenarios. A good model and algorithm can have a high detection rate with low false alarm rate. Therefore, an evaluation curve can be plotted for different system or same system with different parameters given the two parameters.

$$DR = \frac{C_D}{N_C} \times 100\% \quad (7.1)$$

Where:

$DR$ : Detection rate

$C_D$ : The number of conflict events detected

$N_C$ : The total number of conflict events

$$FR = \frac{A_F}{N_A} \times 100\% \quad (7.2)$$

Where:

$FR$ : False alarm rate

$A_F$ : The number of false alarms

$N_A$ : The total number of alarms generated

### 7.1.1 Limitations

The limitation of the traditional approach combining with detection rate and false alarm rate based on conflict events can be used to evaluate the queue warning system such as the proposed one. But, rather than an accurate approach, it is a conservative one. The reason is that CPCs do exist when no conflict event observed. A conflict event usually involves both dangerous conditions and dangerous behaviors of drivers. Crashes can happen under safe conditions even when the traffic condition is safe. By this nature, the detection rate is get underestimated and the false alarm is overestimated. Also, when the warning messages get delivered to the drivers, the traffic conditions during the alarm being on is affected by the queue warning system. Given a perfect queue warning system with perfect impact on the behavior on the drivers, every conflict event should be avoided. In this case, every alarm

will be considered as false. This conflicts with the assumption that such a system is perfect.

A more accurate approach is to use condition-based evaluation methods, in which each condition is labeled as dangerous or not. However, this may not be feasible not only because it may take a long time and huge amount of effort to get sufficient number of labels but also due to the illusiveness and ambiguous in determine dangerous conditions. Since the traffic conditions described is for a zone instead of a single point, makes it hard for manual determinative labeling. Also, judging the level of dangerousness manually can bring personal bias in to a dataset.

An easier way to mitigate the issues with the false alarm rate is to avoid using it since a traffic condition with no conflict event happens does not indicate it is safe and an alarm raised for this condition is false. Therefore, this study use the average time alarm get raised to demonstrate the efficiency of the system and shows the trade off between higher detection rates and longer alarm length.

## 7.2 Performance Evaluating Indexes

### 7.2.1 Event Type and Notation

While traditional evaluation methods classified a event as detected or missed, in the multi-layer system design, a more specific classification of the conflict events can be achieved. Based on the algorithm suggestion, system output and the status of the sign, each conflict event can be put into one of the following categories:

Y1: Driver warned. The sign is up(showing warning) at the time when the conflict event happened.

N1: The system was not in working state(state 0,2,3 as shown in Figure 5.2).

N2: The algorithm does not propose to raise the alarm when the conflict event happened.

N3: The algorithm proposed to raise the alarm but the system output is “AlarmOff” due to congestion override rules.

N4: The algorithm proposed to raise the alarm but the system output is “AlarmOff” due

to time override rules.

N5: The system raised the alarm but the sign hasn't show the alarm yet. Communication delay issue.

Note that all N states refers to situations that the sign was not showing the alarm at the time the event happened.

With these notations, the detection rates for both algorithm level and system level can be found in the following section.

### 7.2.2 Detection Rate

The detection rate is defined in Equation 7.1, the percentage of events that driver were warned in the total number of conflict events. On the system level, such detection rate is calculated using Equation 7.3.

$$Dr_{sys} = \frac{N_{Y1}}{N_{Y1} + N_{N2} + N_{N3} + N_{N4} + N_{N5}} \quad (7.3)$$

Where:

$Dr_{sys}$ : Detection rate on the system level

$N_X$ : The number of conflict events labeled as X, where  $X \in \{Y1, N1, N2, N3, N4, N5\}$

On the algorithm level, such detection rate is calculated using Equation 7.4. The intuition behind this equation is that for the conflict events labeled as  $N3, N4, N5$ , the algorithm provide a suggestion of raising the alarm, but the sign does not deliver the warning to the driver because either override rules or refresh issues. In other words, the algorithm detected these conflict events, but the system did not raise the alarm even if the algorithm suggest that an alarm should be delivered to the drivers. The algorithm detection rate is a upper bound of the system detection rate. Under a ideal implementation environment where the update of the sign is instantaneous, if there are no override rules, the algorithm and system would have the same value in detection rate. By comparing these two detection rates, one can see the impact of override rules and communication issues on the performance of the proposed system.

$$Dr_{alg} = \frac{N_{Y1} + N_{N3} + N_{N4} + N_{N5}}{N_{Y1} + N_{N2} + N_{N3} + N_{N4} + N_{N5}} \quad (7.4)$$

Where:

$Dr_{alg}$ : Detection rate on the algorithm level

$N_X$ : The number of conflict events labeled as X, where  $X \in \{Y1, N1, N2, N3, N4, N5\}$

### 7.2.3 Alarm Length

To measure the efficiency of the system, this study uses alarm length to show how long the alarm has been raised on average for each day. Given the same detection rate, a lower value of alarm length is desired since it is more efficient. Similar to the detection rate, this thesis also calculated the alarm length for both the system level and algorithm level.

For the system level, the alarm length is the total time that the sign shows warning messages to drivers. For algorithm level, the alarm length is the total time that the algorithm suggests that an alarm should be raised. Based on the design of the proposed system, the alarm length of the algorithm level is the upper bound of the alarm length of the system level since in the implemented prototype, system can only deliver a warning when the algorithm suggest so.

## 7.3 The Baseline: A Naive Method

To evaluate the performance of the proposed system, besides comparing the same architecture with different parameters, this study also proposed a naive queue warning algorithm that utilizes two speed threshold to determine the raising and dropping of the alarm. The naive algorithm take the individual vehicle measurement data from downstream detector and determine if an alarm should be raised accordingly. The logic of this method is to setup a pair of speed thresholds triggering threshold,  $f_s$  and ending threshold,  $f_e$  where  $f_s \leq f_e$ . Let  $u$  denote the average speed at downstream detector, then when the alarm is off, a condition that  $u \leq f_s$  will trigger the alarm and the alarm will stay on until a

condition,  $u \geq f_e$  is observed.

Since the naive method does not have the noise-tolerant benefit from the multi-metrics approach, a five minute moving average filter were applied to the individual vehicle speeds. To mimic the proposed system, the warning results are updated every 2 seconds. If no new vehicles arrives in this 2-second interval, the speed will remain the last known value.

To judge the naive method and the proposed method in a fair fashion, the system states(i.e. 0, 1, 2, 3 in Figure 5.2) from the proposed system are used as the system states for the naive method. This is to avoid using the situations that the system was not working in order to handle issues such as poor communication. For this period, it is unknown if the conflict event can be detected by the algorithm/system due to its non-active status.

## 7.4 Evaluation Details

The objective of the evaluation is to compare the the performance of the proposed system with different parameters and also to compare its performance with the baseline model. For the proposed system,one important parameter is the threshold pair  $\{\lambda_1, \lambda_2\}$  that determines the starting and ending of the alarm.

For simplicity, this study choose the pair in a fashion that  $\lambda_1 = \lambda_2$  where  $\lambda_1 = 0.01, 0.02, \dots, 1.00$ . The reason to set  $\lambda_1 = \lambda_2$  is that since  $\lambda_2$  regulates the ending of the alarm, the smaller it gets, the longer an alarm may last. Therefore, when  $\lambda_1 = \lambda_2$ , the system are acting at lower bound of both alarm length and detection rate given certain  $\lambda_1$ .

## 7.5 Performance Comparison

Both the naive method and the off-line prototype get evaluated for the 55 evaluating days from 9:00AM to 9:00PM. Figure 7.1 presents the result of the evaluation. Blue dots represents the performance of the off-line prototype with different parameters, which get connected by a curve to show the trend. The yellow dots represents the performance of the

naive model. The starting speed thresholds  $f_s$  were selected from 0 to 40 with the step of 2. The ending thresholds  $f_e$  were selected in the interval of  $[f_s, 50]$  with the step of 5. The reason for such exhaust selection is to be sure cover as much variants of the naive model as possible.

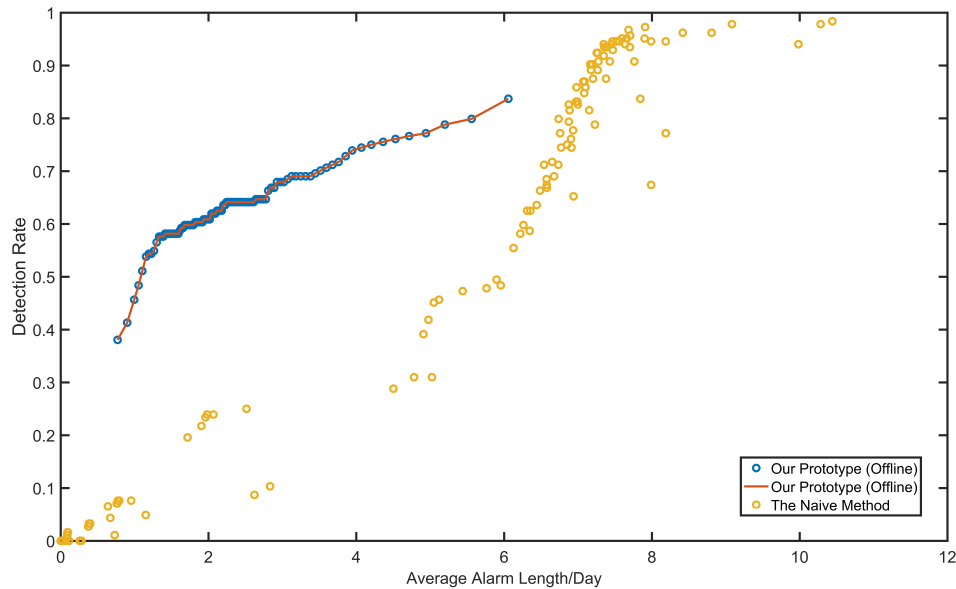


Figure 7.1: Algorithm level evaluation curve(with no override rules)

As we can see in Figure 7.1, our approach started at an overall detection rate a little bit below 40 percent and increase rapidly with the increase of the average alarm length. In this left part of the curve, it shows the high marginal gain of detection rate with the slight increase in average alarm length. This gain, however, reduced rapidly with the increase in average alarm length and were quite steady around the detection rate between 60 to 70 percent. With the lowest value in the tested thresholds, i.e. the right most point on the brown curve where the system has the highest sensitivity, it stopped between 80 and 90 percent, indicating that about 10 to 20 percent of the conflict events didn't get detected with the most sensitive setting tested in this experiment. One hypothesis is that these conflict events may be because of factors other than CPCs.

The performance of the naive method, on the other hand, seems to be quite limited. When the average alarm length is shorter than 4 hours per day (25% of the service time), the detection rate of conflict events are quite low compare to our method. In fact, with raising the alarm for about 25% of the service time, only about 25% of conflict events were successful detected. This is similar to the performance of a random warning generator. There seems to be a linear relationship between the average alarm length and detection rate when the average alarm length is shorter than 6 hours per day, where the performance of the naive method is tie with a random alarm generator. The detection rate increased rapidly when the average alarm length exceeded 6 hours per day and increased to about 100% at about 8 hours per day. One hypothesis for such increase is that with the increase of  $f_s$ , alarms will be produced for almost all traffic conditions. Since one of the extreme threshold tested in this experiment is  $f_s = 40$ , meaning that warnings will be generated for any traffic conditions with a speed lower than 40 mph. And usually, when the speed is higher than 40mph, the traffic is near free-flow without disturbance. Therefore it got almost about 100% of detection rate.

Theses results showed that when the average alarm length is shorter than 6 hours per day, the naive model has a performance similar to random raising the alarm regardless of traffic conditions. Our approach has a much better performance in this region with decent detection rates. In the experiment, the maximum alarm length generated by our prototype is 6 hours per day, which is already 50% of the entire service period.

## Chapter 8

# Result and Discussion

The aim of this Queue Warning system is to efficiently alert drivers of crash-prone conditions they are about to encounter. The threshold for the dangerous level measurement used was among the proposed in [4] for the speed data filter selected. The system operation was monitored during the three phases of the field implementation. Phase 1, March 1st, 2016 to April 30th, 2016 involved the operation of the system in silent mode, meaning the sign was not activated but alarms were recorded. During this phase the system's correct operation and reliability was tested as well as minor adjustments were made in the system parameters. Phase 2, May 1st to June 1st involved the full system operation as well as the introduction of a heuristic test to reduce the time the alarm shuts off at the end of the day when traffic returns to free flow permanently. Phase 3 involved the full operation of the system with no further changes or fine-tuning and is the phase that produced the results presented in this thesis.

The false alarm rate is very difficult to calculate. Unlike the detection rate where the system performance is only evaluated at the time of an event, the only way to compute a false positive or false negative rate would need to be based on some definitive ground truth (the collected video) that determines if conditions were, in fact, crash-prone when the system said they were and calculate the false positive and false negative rates from that. Not only it would be extremely time-consuming to delineate all the times when conditions appeared crash-prone, the task of labeling traffic conditions would be extremely subjective



and therefore, unreliable. As such, no quantitative measure of the systems false alarm rate is available at the moment.

Regardless, given that drivers have shown to lose confidence on other warning systems when the conditions encountered are contrary to the message provided by the system, attention was given in minimizing, if not eliminating, the cases where the system displays a warning but the drivers exposed to it never encounter a reason to stop. Following the fine-tuning that took place in phase 2, such cases have been, to the best of our knowledge, eliminated.

## 8.1 Detection Rates

To assess the performance of the system, the detection rate of all conflict events between 3rd Ave and Chicago Ave during the three-month evaluation period were calculated separately for the control algorithm and for the system as a whole. The detection rate was calculated for just the crashes, near crashes, and both events combined. To find the actual number of conflict events, all the events observed during the evaluation period were tabulated and sorted based on whether the drivers involved were warned or not warned about crash-prone conditions before the event and if not separated by the reason for such failure (the results are tabulated in Table 8.1).

Table 8.1: Breakdown of system performance in right lane by component

Reason For Failure to Warn Driver	Event Type					
	Crashes		Near Crashes		All Events	
	[Count]	[% of Tot.]	[Count]	[% of Tot.]	[Count]	[% of Tot.]
Driver warned	15	36.6%	70	49.0%	85	47.3%
System buffering or inoperative	3	7.3%	8	5.6%	11	6.0%
Alarm was not raised	9	22.0%	28	19.6%	37	20.1%
MnDOT congestion override in place	8	19.5%	10	7.0%	18	9.8%
MnDOT time override in place	4	9.8%	25	17.5%	29	16.8%
System delay	2	4.9%	2	1.4%	4	2.2%
Total	41		143		184	

Using the results shown in Table 8.1, the detection rates were calculated. To assist the reader in comparing the different detection rates and to measure the effect the overrides have on the system performance the results are separated in two categories; one is based on the algorithm decision and the other is based on the whole system which is the algorithm plus the overrides. In addition, these results are grouped into two categories based on time period; one for the full time surveillance data were available (9am to 8pm) and the other for the system operational period of noon to 8pm. The results are tabulated in Table 8.2 which shows the rates at which various event types were detected by the algorithm and the rates at which the system as a whole provided a warning to the drivers involved in those events.

Table 8.2: Detection rates during the trial period

Detecting Component	Event Type		
	Crashes	Near Crashes	Both
	[%]	[%]	[%]
Algorithm (9 a.m. to 8 p.m.)	76.3	79.3	78.6
Whole System (9 a.m. to 8 p.m.)	39.5	51.9	49.0
Algorithm (noon to 8 p.m.)	73.5	74.6	74.3
Whole System (noon to 8 p.m.)	44.1	63.6	59.2

It is apparent that the control algorithm is far more successful at raising the alarm for events than the system as a whole. The noon to 8pm group illustrated the effect of the congestion override, a subject we have raised as a concern to MnDOT. As seen in Table 8.1, following a successful detection by the control algorithm, the main reason the sign was not activated is the presence of a MnDOT override.

Table 8.3: Crash frequencies per million vehicles

Time Period	2013 Crash Frequency		2016 Crash Frequency	
	Frequency	C.I.	Frequency	C.I.
All Day	19.43	[11.96,26.90]	26.08	[17.32,34.85]
11 AM to 7 PM	40.27	[24.49,56.06]	44.51	[27.40,61.62]
12 PM to 7 PM	46.11	[28.03,64.18]	43.33	[25.22,61.43]

Table 8.4: Near crash frequencies per million vehicles

Time Period	2013 Near Crash Frequency		2016 Near Crash Frequency	
	Frequency	C.I.	Frequency	C.I.
All Day	192.82	[169.29,216.35]	109.71	[91.72,127.69]
11 AM to 7 PM	393.06	[343.74,442.38]	195.17	[159.34,231.00]
12 PM to 7 PM	403.90	[350.41,457.40]	187.10	[149.48,224.73]

Table 8.5: Conflict event frequencies per million vehicles

Time Period	2013 Conflict Event Frequency		2016 Conflict Event Frequency	
	Frequency	C.I.	Frequency	C.I.
All Day	212.25	[187.57,236.94]	135.79	[115.78,155.79]
11 AM to 7 PM	433.33	[381.55,485.12]	239.68	[199.98,279.38]
12 PM to 7 PM	450.01	[393.55,506.48]	230.43	[188.68,272.18]

Table 8.6: Event frequency comparison

	Crash	Near Crash	All Events
All Day	34.23%	-43.11%	-36.02%
11 AM to 7 PM	10.53%	-50.35%	-44.69%
12 PM to 7 PM	-6.03%	-53.68%	-48.79%

Table 8.3 and Table 8.4 show the frequencies at right lane on weekdays in June, July, and August of 2013 and 2016. As discussed before, some of the days in 2016 were not used due to the lack of data, some days with similar dates in 2013 were not used to make sure two periods have the same number of days. Also, the camera has a larger coverage zone in 2016 and also a longer coverage period compared to 2013. Conflict events happen at locations covered in both 2013 and 2016 were used for this comparison. In both of these two tables, all day crash rates may not be a fair comparison since 2016 has longer coverage compared to 2013.

Table 8.3 shows the crash frequency in these two years. During the period 11 AM to 7 PM, for which the conflict event in both years were collected, there's a slight increase in crash frequency. When looking at the 12 PM to 7PM(the system only provided warning after 12 PM due to time override), there's a slight decrease in crash frequency. However, since the 95% confidence intervals are wide and highly overlapped, it may suggest that the it is not statistical significant to compare the performance of the crash frequency in these two years. It remains a puzzle of the impact of the system on crash frequencies.

Since crashes are rare events, it is hard to draw a reliable conclusion of the impact of the proposed system. However, near crashes, which has a larger sample size, may help to complete this task. As Table 8.4 shown, the frequency of near crashes get reduced with no overlapping of the 95% confidence intervals. Table 8.5 shows the frequency of all conflict events before and after the deployment of the system. Table 8.6 shows the comparison of the conflict events between 2013 and 2016. As the table shows, there's about 50% of reduction in near crashes from 11AM to 7PM and similar reduction in the frequency of all conflict events. Unfortunately, without the assumption that the impact of the system on both conflict events is similar, the result could not yield a reliable conclusion for the system's ability in crash reduction.

To compare the traffic conditions in these two years, Figure 8.1 presents the histograms for the occupancy data for the right lane at both upstream and down stream detectors in the year 2013 and 2016. The occupancy for these two years have almost identical distributions which indicates the differences in the number of conflict events were not because the change in traffic conditions.

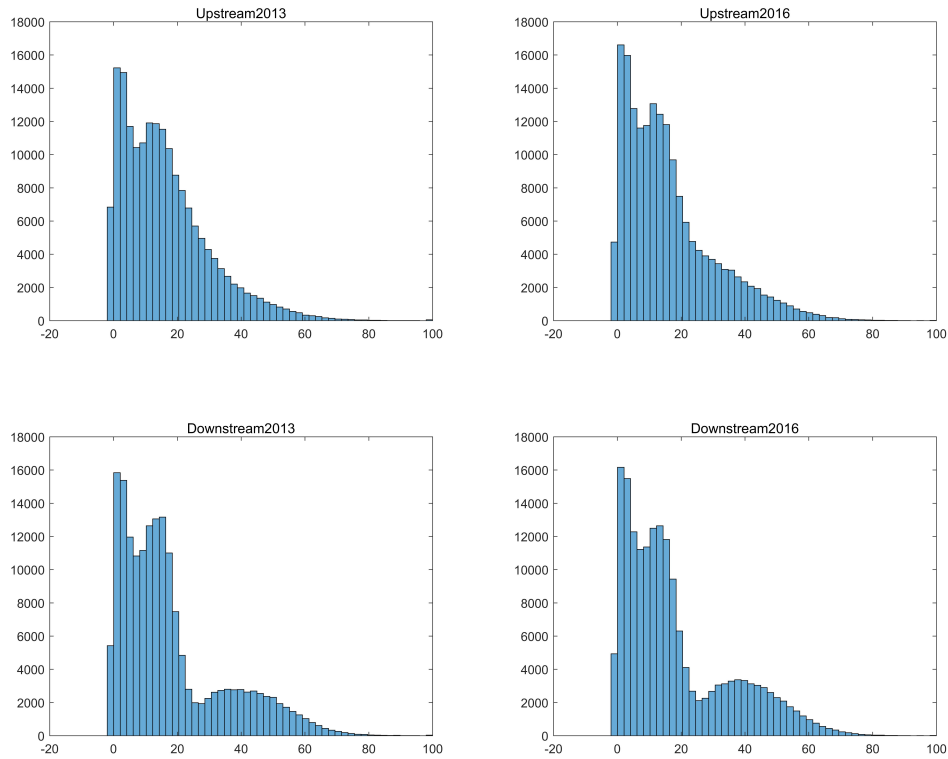


Figure 8.1: Occupancy Comparison

## 8.2 System Limitations

Due to the fact that the system is still in the prototype phase, several aspects of it are less than ideal and result in uncertainties of varying degrees. The limitations and uncertainties include the non-ideal configurations of the MVDs and warning sign, the limitations of the hardware, the case that some of the equipment was shared with MnDOT, and the actuality that the model result is currently computed for only the right lane. In the interest of cost and time, the MVDs were installed on a high-rise near the road rather than on a gantry or bridge directly over the road. The warning messages were displayed on the relatively small MnDOT changeable message board rather than a purpose-made display with potentially

different characteristics (location, size, color, etc.). Also, current policy does not permit the free choice such as different messages and colors despite that the sign can display different color and different messages. Warning messages can be classified by informative messages (information about the traffic flow such as “slow traffic ahead”) and suggestive warnings (suggestions to drivers such as “prepare to stop”). Having more than one signs deployed on the freeway segment and first presents informative messages at upstream and then presents the suggestive messages near the detection zone may achieve a better result and a higher performance. Unfortunately, these experiments were not possible due to the policies at the time this thesis was composed. The limitations of the hardware also presented a source of uncertainty in that the speed data from the MnDOT single loop detector was only a 30-second average and was not completely accurate. Issues with power outages and lack of sufficient data storage at the rooftop station resulted in missing video data. Because control of the sign and the camera monitoring was shared with MnDOT, the sign was occasionally used to display incident management messages from MnDOT rather than the queue warning messages. Since the capability of the system to function as intended and the effect that it would have on drivers were unknown at the time of installation, it was only applied to one of the three lanes. As a result, it is unclear what effect this partial deployment had on driver’s actions.

## Chapter 9

# Conclusion and Future Studies

### 9.1 Conclusion

Impressions from the evaluation period suggest that the Queue-Warning system prototype discussed in this document represents a successful crash prevention solution. The effect of the system on conflict events is difficult to determine from the data available but the system performs well given the constraints under which it is operating. The system raised the alarm for 79% of all conflict events that occurred during the trial period but was only able to warn drivers for 49% of all conflict events predominantly due to the overrides that were put in place to prevent the possibility of the system displaying the warning for excessive amounts of time. Why these crashes happened even with the drivers exposed to the warning is unknown. One can speculate that the message is not clear or strong enough. A lot of the vehicles involved are only exposed to the warning once on the Park Ave gantry since the earlier one is not visible from the entrance from 35W SB.

In conclusion, this thesis presents the design, specification, implementation and evaluation of an infrastructure based Queue Warning system that is capable of detecting crash-prone traffic conditions, on freeways and delivering warning messages to drivers in order to increase their alertness to these traffic conditions and ultimately reduce crash frequency. A multi-layer system design, for a robust and stable real-time queue warning system, was developed and implemented in a real-world context based on sensors providing individual



vehicle measurements. The details of a prototype system was presented and different parameters of this prototype were tested. By comparing the performance of the prototype with the performance of a naive method, this thesis illustrates the higher performance and efficiency of the proposed methodology. The prototype was deployed on a high crash frequency freeway section at the right lane of a 1.7-mile-long freeway segment of Interstate 94 Westbound near downtown Minneapolis. A three-month investigation of the operations of the QWarn system showed the event frequency reduced from 212.25 to 135.79 conflict events per million vehicle traveled. With huge amount of effort and time devoted in designing, coding, data analysis, system evaluation and testing, this work filled the gap between theoretical works and practical uses. The framework proposed by this study allows other methodologies modeling dangerousness level of traffic conditions to be applied in the dangerous level measurement layer.

## 9.2 Future Studies

This work focused mainly on proof-of-concept in a field operational test environment, the next step is to expand and optimize the infrastructure used by the system as seek improvements on the algorithm. Several changes to the system at the policy and infrastructure levels have been identified that will likely result in greater detection and warning rates. Given greater design control, many of the system limitations could be mitigated if not eliminated altogether. This will include installing MVDs or other sensors on gantries or bridges directly above the freeway in so they are afforded a stable head-on view of traffic and placing the warning sign in a location selected by choice rather than necessity. The warning sign shape, size, location, and message can also be optimized to increase driver attention and compliance. The most important improvement will be expanding the system to all three lanes of traffic as well as conduct a long-term study of the effects of the system on conflict event frequencies.

For the methodology level, future studies may focus on two problems: traffic condition labeling and a better measurement of condition dangerousness. As crash prone conditions do not always result in a crash, labeling traffic conditions where no crash happened as safe

may not be accurate. Models trained or fitted based on these inaccurate labeled data may regard these crash prone conditions as safe or get confused by these conditions. To get better labeling, many new approaches can be applied. For instance, unsupervised machine learning and data analysis techniques such as clustering can be applied to form condition clusters. Their correlation with crashes can be used to discover the dangerous clusters in which many conditions may not associated with any conflict events. To get a better measurement of condition dangerousness, many supervised learning techniques such as deep constitutional neural networks can be applied. In the field of image recognition, deep constitutional neural networks have been yielding very decent results in the past several years. Applying and re-innovating the cutting-edge methodologies from the machine learning field may produce much accurate model for measuring condition dangerousness and therefore, make it possible to improve the performance of the proposed system. Besides condition labeling and dangerousness measurement, future study can also focus on uncovering universal patterns that is invariant of the type of network, i.e. can be directly applied on any type of road networks besides freeway. This may be achieved by domain adaptation techniques such as domain adversarial learning. Reinforcement learning techniques, such as deep Q learning, may lead to a system that is able to improve its accuracy and performance during its operation. In a word, the extension and future study based on this thesis are in a wide range and may not only improve the performance of the proposed system, but also brought intellectual merit and a broad impact in the field of transportation.

# References

- [1] Early Estimate of Motor Vehicle Traffic Fatalities in 2014, <https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812160>
- [2] Benefit-Cost Analysis for Transportation Projects, [http://www.dot.state.mn.us/planning/program/appendix\\_a.html](http://www.dot.state.mn.us/planning/program/appendix_a.html)
- [3] Minnesota Traffic Crashes in 2014, <https://dps.mn.gov/divisions/ots/reports-statistics/Documents/2014-crash-facts.pdf>
- [4] Hourdos, J., Garg, V., Michalopoulos, P., & Davis, G. (2006). Real-Time Detection of Crash-Prone Conditions at Freeway High-Crash Locations. *Transportation Research Record*, 1968(1), 8391. doi: 10.3141/1968-10
- [5] Qiu, L., & Nixon, W. A. (2008). Effects of Adverse Weather on Traffic Crashes: Systematic Review and Meta-Analysis. *Transportation Research Board*, 2055, 139146. doi:10.3141/2055-16
- [6] Kopelias, P., Papadimitriou, F., Papandreou, K., & Prevedouros, P. (2007). Urban Freeway Crash Analysis: Geometric, Operational, and Weather Effects on Crash Number and Severity. *Transportation Research Record: Journal of the Transportation Research Board*, 2015, 123131. doi:10.3141/2015-14
- [7] Golob, T. F., & Recker, W. W. (2003). Relationships Among Urban Freeway Accidents, Traffic Flow, Weather, and Lighting Conditions. *Journal of Transportation Engineering*, 129(August), 342353. doi:10.1061/(ASCE)0733-947X (2003)129:4(342)

- [8] Zheng, Z., Ahn, S., & Monsere, C. M. (2010). Impact of traffic oscillations on freeway crash occurrences. *Accident Analysis and Prevention*, 42(2), 626636. doi: 10.1016/j.aap.2009.10.009
- [9] Abdel-Aty, M., Uddin, N., Pande, A., Abdalla, F., & Hsia, L. (2004). Predicting Freeway Crashes from Loop Detector Data by Matched Case-Control Logistic Regression. *Transportation Research Record*.
- [10] Caliendo, C., Guida, M., & Parisi, A. (2007). A crash-prediction model for multilane roads. *Accident Analysis and Prevention*, 39(4), 657670. doi: 10.1016/j.aap.2006.10.012
- [11] Pei, X., Wong, S. C., & Sze, N. N. (2011). A joint-probability approach to crash prediction models. *Accident Analysis and Prevention*, 43(3), 11601166. doi: 10.1016/j.aap.2010.12.026
- [12] Naderan, A., & Shahi, J. (2010). Aggregate crash prediction models: Introducing crash generation concept. *Accident Analysis and Prevention*, 42(1), 339346. doi: 10.1016/j.aap.2009.08.020
- [13] Montella, A., Colantuoni, L., & Lamberti, R. (2008). Crash Prediction Models for Rural Motorways. *Transportation Research Record: Journal of the Transportation Research Board*, 2083, 180189. doi:10.3141/2083-21
- [14] Abdel-Aty, M. A., & Pemmanaboina, R. (2006). Calibrating a real-time traffic crash-prediction model using archived weather and ITS traffic data. *IEEE Transactions on Intelligent Transportation Systems*, 7(2), 167174.
- [15] Abdel-Aty, M., & Gayah, V. (2010). Real-Time Crash Risk Reduction on Freeways Using Coordinated and Uncoordinated Ramp Metering Approaches. *Journal of Transportation Engineering*.
- [16] Lee, C., Hellinga, B., & Saccomanno, F. (2003). Real-Time Crash Prediction Model for Application to Crash Prevention in Freeway Traffic. *Transportation Research Record: Journal of the Transportation Research Board*, 1840, 6777. doi:10.3141/1840-08

- [17] Khan, A. Intelligent infrastructure-based queue-end warning system for avoiding rear impacts. *IET Intell. Transp. Syst.*, Vol. 1, No. 2, 2007, p. 138.
- [18] ARTBA Work Zone Safety Consortium. Innovative End-of Queue Warning System Reduces Crashes Up to 45
- [19] Pesti, G., P. Wiles, R. Cheu, P. Songchitruska, J. Shelton, and S. Cooner. Traffic Control Strategies for Congested Freeways and Work Zones. Texas Department of Transportation, 2007.
- [20] Delta Highway Queue Warning System Evaluation - Final Technical Memorandum. <http://www.kittelton.com/projects/intelligent-transportation-systems-its-statewide-evaluations>. Accessed Jul. 19, 2016.
- [21] Mahmassani, H., Rakha, H., Hubbard, E., and Lukasik, D. Concept Development and Needs Identification for Intelligent Network Flow Optimization (INFLO). USDOT ITS Joint Program Office, 2012.
- [22] Hourdakis, J., Michalopoulos, P. G., and Morris, T. Deployment of Wireless Mobile Detection and Surveillance for Data-Intensive Applications. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 1900, Transportation Research Board of the National Academies, Washington, D.C., 2004, pp. 140148.
- [23] Jones, Trevor R., and Potts Renfrey B. "The Measurement of Acceleration Noise-A Traffic Parameter." *Operations Research* 10.6 (1962): 745-63. Web.
- [24] Helly, W., and Baker, P.G., (1965). Acceleration noise in a congested signalized environment. *Vehicular Traffic Science. Proceedings of the Third International Symposium on the Theory of Traffic Flow*. American Elsevier, New York. pp. 55-61.
- [25] Greenshields, B. D. (1961) *Quality of Traffic Flow, Quality and Theory of Traffic Flow*. Symposium, Bureau of Highway Traffic, Yale University, New Haven, Conn. pp.3-40.
- [26] Phillips, W. F., (1979). A kinetic model for traffic flow with continuum implications. *Transportation Planning and Technology* 5, 131-138.
- [27] Drew, Donald. R, (1968) *Traffic Flow Control and Theory*, McGraw Hill.

- [28] Hourdos, J. (2005) Crash prone traffic flow dynamics: Identification and real-time detection. Ph.D. dissertation, University of Minnesota, 315 pages.
- [29] Subramaniam V. K. (2001) Detection of Undesirable Traffic Flow Conditions. M.S. Thesis, University of Minnesota.