

A FRAMEWORK FOR SIMULATING VARIABLE SPEED LIMIT ALGORITHMS IN
CORSIM

By

CLARK LETTER

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2011

© 2011 Clark Letter

To my family and friends

ACKNOWLEDGMENTS

I thank my advisor, Dr. Lily Elefteriadou, Professor, Department of Civil and Coastal Engineering, University of Florida, for her guidance and support throughout the duration of my thesis. I would also like to thank my committee members, Dr. Scott Washburn, Associate Professor, and Dr. Yafeng Yin, Assistant Professor, for their guidance and feedback on the study. I would like to give a special thanks to Mr. Tom Simmerman, for his assistance in computer programming and understanding of internal Corridor Simulation (CORSIM) structures. Finally, I would like to thank my family and friends for their constant moral support and encouragement.

TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS	4
LIST OF TABLES	7
LIST OF FIGURES	8
ABSTRACT.....	10
CHAPTER	
1 INTRODUCTION	12
1.1 Background.....	12
1.2 Objectives	13
1.3 Thesis Outline.....	13
2 LITERATURE REVIEW	14
2.1 Implementation of variable speed limits (VSL)	14
2.1.1 Implementation in the USA.....	14
2.1.2 Implementation in Europe	19
2.2 VSL Simulation	24
2.3 VSL Algorithms.....	32
2.3.1 Congestion and Safety-Related Algorithms	32
2.3.2 Weather-Related and Other Algorithms	34
2.4 Corridor Simulation (CORSIM) Simulation Software	35
2.5 Literature Review Summary	36
3 METHODOLOGY	39
3.1 Study Site Selection and Data Assembly.....	39
3.2 Development of run time extensions (RTE).....	39
3.3 Testing and Analysis of Algorithms in CORSIM.....	40
4 STUDY SITE DESCRIPTION.....	41
4.1 Study Site Selection	41
4.2 Calibrated CORSIM Simulation.....	42
5 DEVELOPMENT OF THE RTE	47
5.1 Selection of Algorithms.....	47
5.1.1 Algorithm Based on Occupancy.....	48
5.1.2 Algorithm Based on Flow	48

5.1.3 Algorithm Based on a Logic Tree including Flow, Occupancy, and Average Speed.....	49
5.2 Sign Location Variations	49
5.3 Building of dynamic link libraries (DLLs) to communicate with the RTE interface.....	50
6 IMPLEMENTATION AND ANALYSIS OF ALGORITHMS IN CORSIM	61
6.1 Implementation of RTE Scenarios in CORSIM	61
6.2 Analysis of the No Control Scenario	62
6.3 Analysis of VSL Algorithm Performance	62
6.3.1 Analysis of the Occupancy-Based Algorithm	63
6.3.2 Analysis of the Volume-Based Algorithm	64
6.3.3 Analysis of the Multiple Parameter-Based Algorithm	66
6.4 Summary of Analysis	67
7 SUMMARY AND CONCLUSIONS	111
APPENDIX	
A CALIBRATION RESULTS FOR I-95 NETWORK	114
B Sample Source Code for RTE.....	126
C Speed Profiles for scenarios tested	134
LIST OF REFERENCES	182
BIOGRAPHICAL SKETCH	185

LIST OF TABLES

<u>Table</u>		<u>page</u>
2-1	Orlando, Florida I-4 variable speed limit (VSL) control thresholds.....	37
5-1	Occupancy thresholds for displayed speed limits (scenario 1).....	52
5-2	Occupancy Thresholds for Displayed Speed Limits (scenario 2).....	52
5-3	Occupancy thresholds for displayed speed limits (scenario 3).....	52
5-4	Volume thresholds for displayed speed limits (scenario 1).....	53
5-5	Volume thresholds for displayed speed limits (scenario 2).....	53
6-1	Description of scenarios tested	70
6-2	Network performance measures for the occupancy-based scenario	71
6-3.	Occupancy thresholds for displayed speed limits (scenario 2).....	72
6-4	Network performance measures for the volume-based algorithm	73
6-5	Network performance measures for the multiple parameter-based algorithm.....	73

LIST OF FIGURES

<u>Figure</u>	<u>page</u>
2-1 Decision path for determining the new posted speed of the trigger variable speed limits (VSLs).....	38
2-2 Operation of Corridor Simulation (CORSIM) and run time extensions (RTEs)	38
4-1 Section of I-95 being simulated by CORSIM.....	43
4-2 Location of detectors along the section of I-95 study site containing proposed VSL implementation	44
4-3 Speed profile of the I-95 section for time period 4.....	45
4-4 Screen shot of I-95 network represented in CORSIM	46
5-1 Decision tree logic for third algorithm.....	54
5-2 Use of one sign spaced approximately one-half mile	55
5-3 Use of one sign spaced approximately one mile.....	55
5-4 Use of two signs spaced approximately one-half mile apart	56
5-5 Use of two signs spaced approximately one mile apart.....	56
5-6 Screen shot of text outputting to screen when speed changes	57
5-7 Flowchart of general RTE logic.....	58
5-8 Time series of speed limit propagation downstream from VSL sign location	59
6-1 Setting up the RTE tool configuration	74
6-2 Specifying the call points of the RTE.....	75
6-3 Speed profile for no-control scenario over time the 12 periods.....	76
6-4 Speed profile for the occupancy-based algorithm using two signs spaced ½ mile apart (threshold scenario 2) compared to the no-control scenario.....	80
6-5 Throughput for the occupancy-based algorithm using two signs spaced ½ mile apart (threshold scenario 2) compared to the no-control scenario.....	84
6-6 Speed profile for the volume-based algorithm using one sign spaced 1 mile from the bottleneck source (threshold scenario 1) compared to the no-control scenario.....	85

6-7	Speed profile for the volume-based algorithm using two signs spaced 1/2 mile apart (threshold scenario 1) compared to the no-control scenario	89
6-8	Throughput for the volume-based algorithm using one sign spaced 1 mile from the bottleneck source (threshold scenario 1) compared to the no-control scenario.....	93
6-9	Throughput for the volume-based algorithm using two signs spaced 1/2 mile apart (threshold scenario 1) compared to the no-control scenario.....	94
6-10	Logic tree thresholds used for the multiple parameter algorithm	95
6-11	Speed profile for the multiple parameter--based algorithm using two signs spaced 1/2 mile apart compared to the no-control scenario.....	96
6-12	Speed profile for the multiple parameter-based algorithm using one sign spaced 1/2 mile from the bottleneck source compared to the no-control scenario.....	100
6-13	Throughput for the multiple-parameter-based algorithm using two signs spaced 1/2 mile apart compared to the no-control scenario.....	104
6-14	Throughput for the multiple-parameter-based algorithm using one sign spaced 1/2 mile from the bottleneck source compared to the no-control scenario.....	105
6-15	Speed profile for the multiple parameter-based algorithm using one sign spaced 1 mile from the bottleneck source compared to the no-control scenario.....	106
6-16	Throughput for the multiple-parameter-based algorithm using one sign spaced 1 mile from the bottleneck source compared to the no-control scenario	110

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

A FRAMEWORK FOR SIMULATING VARIABLE SPEED LIMIT ALGORITHMS IN
CORSIM

By

Clark Letter

August 2011

Chair: Lily Elefteriadou
Cochair: Scott Washburn
Major: Civil Engineering

A major problem associated with freeway operations around major cities is congestion occurrence during peak volume periods. Typically, bottlenecks at merging and diverging junctions as well as incidents create a shockwave that propagates upstream. One of the tools currently examined as a way to dampen the shockwave produced by this bottleneck is variable speed limits (VSL). Current micro-simulators do not provide an interface to easily simulate VSLs and evaluate their impact on traffic, thus simulation must be carried out through additional coding. This study creates a test-bed for simulating and evaluating multiple VSL algorithms using the Corridor Simulation (CORSIM) micro-simulator. Three algorithms for VSL control are selected and simulated to evaluate the effectiveness of each algorithm. The roadway used for the simulation is a 13-mile section of I-95 in Miami, Florida. A run-time extension (RTE) interface is built to communicate with the CORSIM simulation and replicate the VSL operations. Different threshold values are tested to evaluate the effectiveness of each algorithm under various settings. It was concluded that all but one of the scenarios tested show an improvement in the average travel speed and total travel time after VSL is implemented. The throughput for

most scenarios showed an improvement when observed over the time duration of the congestion.

Overall, the volume-based algorithm showed the most improvement in the simulations.

CHAPTER 1 INTRODUCTION

1.1 Background

Static speed limits are designed to provide motorists with a safe speed at which to drive. While these safe speeds are effective during ideal conditions, they fail to provide recommended safe speeds during adverse weather or congested driving conditions (Sisiopiku 2001). Variable speed limits (VSLs) are a way of recommending safe driving speeds during less than ideal conditions. VSL systems have produced safety benefits such as a reduced number of rear end collisions and traffic homogenization.

In addition to their safety benefits, VSL have been used upstream of bottlenecks with recurring congestion as a way to dampen the shockwave produced once congestion starts. However, the exact effects of VSL on traffic flow are not well understood, and the literature provides conflicting conclusions with respect to the effectiveness of VSL installations in increasing overall speeds and throughput. With growing interest in identifying and implementing congestion mitigation techniques, there is a need to simulate VSL and evaluate their potential impacts on traffic conditions in a more comprehensive manner. Simulation is a very effective tool in evaluating alternatives under completely controlled conditions which cannot be achieved in the field. It is also very effective in providing a comprehensive picture of traffic operations in time and space.

To-date, few micro-simulators possess the ability to simulate VSL systems. Simulators such as AIMSUN and PARAMICS have the ability to simulate variable message signs, but require additional coding to simulate VSLs. No micro-simulator has a built-in interface that allows the simulation of different VSL systems or algorithms. There are few tools or guidelines available for simulating VSLs, which makes simulating them a difficult and time-consuming

process. Corridor Simulation (CORSIM) is a widely used micro-simulator that does not have an interface to directly simulate VSLs. CORSIM has a run time extension (RTE) interface that allows users to define or modify operations of the simulation program. This interface can be used to simulate VSL operations and to test their effectiveness under a variety of conditions and algorithm settings

1.2 Objectives

The objective of this research is to develop a framework for simulating and testing VSL algorithms in CORSIM through the use of an RTE. Three different algorithms for VSL control are implemented into a CORSIM network and evaluated based on selected performance measures. The network analyzed is a 13-mile section of I-95 in Miami, Florida. This section was selected because the Florida Department of Transportation (FDOT) is interested in evaluating whether VSL would be an effective strategy along this corridor. Different threshold values as well as several different VSL sign locations are tested with each algorithm to evaluate its effectiveness under different settings.

1.3 Thesis Outline

The remainder of this thesis is organized in 6 chapters. In Chapter 2, a literature review on the history of VSL operations both implemented and simulated VSL systems is presented. Chapter 3 discusses the methodology for carrying out the research. Chapter 4 presents the details of the selected study site and the CORSIM network. Chapter 5 explains the development of the RTE and the different algorithms tested. Chapter 6 presents the detailed procedures for performing the simulations, along with the analysis of results. Finally Chapter 7 provides overall conclusions, and identifies directions for future research.

CHAPTER 2 LITERATURE REVIEW

This literature review provides a summary of variable speed limit (VSL) implementation in the United States, as well as in Europe. It also describes how VSL algorithms have been simulated using different software packages, enabling new and creative algorithms to be tested. Next, a summary of different algorithms is presented. Finally, an overview of the Corridor Simulation (CORSIM) software is presented, along with a discussion of the run time extension (RTE) function and its communication with CORSIM.

2.1 Implementation of VSL

This section provides an overview of implemented VSL systems. The section starts with a review of systems implemented in the United States. The section then reviews implementations in Europe.

2.1.1 Implementation in the USA

In the United States variable speed limits have been implemented in a number of locations. These systems typically set a safety speed limit according to the weather, traffic, or road conditions (McLawhorn, 2003). One of the most general uses of VSLs are at school zones and at construction or work zones (Hines, 2002). The main objective of most freeway implementations in the US has been to improve safety, but few have focused on relief of traffic congestion. Congestion-related benefits have been shown mostly using simulation. However, safety benefits have been documented for numerous VSL systems implemented in the field Abdel-Aty et al., (2006).

The first VSL system in the US was implemented along the M-10 (Lodge Freeway) in Detroit, Michigan, between the Edsel Ford Freeway (I-94) and the Davison Freeway in 1960. The system was designed to alert motorists to slow down when approaching congestion and

accelerate when leaving a congested area. The system was 3.2 miles long and had 21 VSL sign locations. The speed limits were chosen by the operator based on closed circuit television and plots of freeway speed. The VSL signs were manually switched at the control center with an increment of 5 miles per hour (mph) from 20 to 60 mph. The evaluation results show that the VSL system did not significantly increase or decrease vehicle speeds (Robinson, 2000). The system was disbanded sometime after 1967.

In New Jersey, a VSL system was implemented along the New Jersey Turnpike in the 1960s. This system was designed to reduce speed limits during congested conditions, and is currently part of a larger ITS system. The system warns drivers of lane closures and crashes to improve safety and avoid large delays. The system is over 148 miles in length and utilizes approximately 120 signs. Since the implementation of the system there have been updates to controllers and detectors to keep the system up to date and functioning effectively. The posted speed limits are based on average travel speeds and are displayed automatically. The posted speed limit can be reduced from the normal posted speed limit (65 mph, 55 mph, or 50 mph) in increments of 5 mph to a minimum speed of 30 mph under six conditions: vehicle collisions, traffic congestion, construction, icy road conditions, snowfall, and fog. No formal evaluation of the system has been performed, but the Turnpike Authority observes the system 24 hours a day and has deemed its performance to be satisfactory. They did note that the system needed enforcement by State Police (Steel et al., 2005).

In New Mexico, a VSL system was implemented along I-40 in Albuquerque in March of 1989 (Robinson, 2002). The system was set up as a test-bed for VSL equipment and was disbanded in 1997 due to road widening. The six kilometer-long system used three roadside detector stations, and a variable message sign to vary the posted speed limit. The posted speed

limit was generated using a look-up table based on the smoothed (90 percent old data plus 10 percent current data) average speed plus, a constant based on the environmental conditions. The speed and environmental data such as light level and precipitation were collected by detectors. Evaluation results show that there was a slight reduction in accidents after the system was implemented. It has been suggested that the implementation of the National Maximum Speed Limit (55 mph) hindered the effect of the system, as posted speeds were generated based on older data, and field conditions didn't match the expected conditions (Steel et al., 2005).

In Tennessee, a VSL system was implemented along a 19-mile section of I-75 in 1993 to respond to the reduction in visibility causing crashes during adverse weather conditions (especially fog) (Robinson, 2002). The system has 10 VSL signs, 8 fog detectors, 44 radar speed detectors, highway advisory radio, and 6 swinging gates. The posted message and speed limit are determined by a central computer in the Highway Patrol office, based on the transmitted data collected using environmental sensor and vehicle detectors. The system has the capability to close down the entire stretch of roadway during severe fog conditions, and divert traffic onto US Highway 11. This requires coordination with highway patrol officers closing swinging gates. The effect of the VSL on actual travel speeds has not been formally evaluated, but the enforcement agency observed a slight (5 to 10 percent) reduction in speed, and there have been no crashes due to fog after the system was implemented. (Goodwin, 2003; Steel et al., 2005).

In Colorado, a VSL system was implemented along the Eisenhower Tunnel on I-70 west of Denver in 1995. This system is designed to improve truck safety by displaying vehicle-specific safe operating speeds for long downgrades. The system consists of a weigh-in motion sensor, variable message sign, inductive loop detectors, and computer hardware and software. A safe speed is computed by an algorithm within the computer system based on the truck weight, speed,

and axle configuration. The recommended safe speed is then displayed on a variable message sign. Moreover, each truck receives a vehicle-specific recommended safe speed message. The speed limit was advisory and evaluation results show that truck-related accidents declined on the steep downhill grade sections after the implementation of the VSL system, even though the truck volume increased (Robinson, 2000).

The Washington Department of Transportation implemented a VSL system on I-90 across the Snoqualmie Pass in 1997. The system was implemented to improve safety and inform motorists of road conditions and weather information and is still active. Speed limits are recommended by the central computer based on information collected from a variety of sources, including wide aperture radar that tracks speeds, roadside cabinets that collect and control roadside data, and packetized data radio on three mountaintop relay sites that use microwaves to communicate to the control center. The computer automatically computes the speed from relayed data and recommends a VSL value, which an operator implements. It was found that VSLs may lose their effectiveness without enforcement by the State Patrol, and they reduced the mean speed and increased the speed standard deviation (Goodwin, 2003; Ulfarsson et al., 2001; Steel et al., 2005).

In 1998, Northern Arizona University and the Arizona Department of Transportation developed a VSL system based on a fuzzy control algorithm along the I-40 corridor in rural Arizona. This was an experimental system designed to display appropriate speeds for different weather conditions. It was unclear from the study whether the system was actually ever implemented, or just simulated. The system used a Road Weather Information System to gather atmospheric and road surface conditions. The system then displayed a corresponding speed limit according to the fuzzy control algorithm. Placer (2001) summarized upgrades made to this Road

Weather Information System. No performance measures or quantitative impacts of the VSL system were given.

In 2000 a VSL system was implemented along I-80 in Nevada. The system was remotely controlled without human intervention. It consisted of four VSL signs (two eastbound and two westbound), visibility detectors, speed loops, RWIS weather stations, and “reduced speed ahead when flashing” signs upstream of the VSL signs. Speed limits were updated every 15 minutes and computed using a logic tree based on the 85th percentile speed, visibility, and pavement conditions. The results found that the sensors were unreliable and could not accurately relay visibility conditions (Robinson, 2000; Robinson, 2002). This limited the effectiveness of the VSL system. No information was found on the current operational status of the system.

In Florida, VMS were placed along a 9-mile portion of I-4 in Orlando. The system is designed to improve safety along I-4 through more steady flow during congested periods, and to provide advance warning of slowing traffic ahead. Detectors are used to measure speed, volume, and occupancy for each lane at 30-second intervals. The SunGuide software monitors the occupancy level and classifies traffic conditions as either free-flow, light congestion, or heavy congestion. On the basis of these classifications, the software recommends speed limits of 30 mph for heavy congestion, 40 mph for light congestion, and the normal speed limit (i.e., 50 or 55 mph) for free flow. The software also ensures that the posted speed limit does not change by more than 10 mph between two adjacent sets of VSL signs (Haas et al., 2009). A study prepared for the FDOT evaluated the performance of the current VSL operation (PBS&J, 2009). The study concluded that the VSL system was not effective at reducing vehicle speeds. Since vehicles were not affected by the signs no traffic improvements or safety benefits were shown.

A study was conducted in southeast Wyoming (Buddemeyer, 2010) to assess the effectiveness of VSL signs in a rural setting on a 100 mile stretch of I-80 through Elk Mountain. The system is designed to reduce speed limits during adverse weather conditions. When a reduced speed limit is in effect, a yellow flashing light on top of the sign is activated and a reduced speed message is displayed. The study showed that vehicle speeds were reduced by 0.47 to 0.75 mph for every 1 mph reduction in posted speed.

In Seattle, Washington VSLs have been installed recently on a stretch of I-5 from Boeing access road to I-90. The project began in 2009 with the installation of fifteen new overhead sign bridges. The system was activated in August 2010. The overhead signs feature individual displays for each lane and warn of approaching lane closures and traffic congestion. The project is designed to reduce the number of collisions and collision-related congestion. The displayed speed limit ranges from 40 mph to 60 mph, and is based on speed and volume data. The speed limit is enforced by the Washington State Patrol. There has yet to be a formal assessment of the effectiveness of the system (WSDOT, 2010).

2.1.2 Implementation in Europe

According to Hines (2002), numerous VSL systems have been implemented in European countries. Based on European case studies, he reported that VSLs can stabilize traffic flow in congestion and thus decrease the probability of crashes. The following provides an overview of VSL implementation in Europe.

A VSL system was implemented along an 18-km (11-mile) section of Autobahn 9 near Munich, Germany, in the 1970s. The system was originally implemented to improve safety, but the effects of the VSL system on other key parameters were also evaluated. The system displays speeds based on three control strategies: incident detection, harmonization, and weather conditions. Boice et al. (2006) investigated the effects of the system on key parameters around

bottleneck formation, based on one-day data along the site. It was found that once a bottleneck had formed there was an 11% reduction in flow in the northbound direction and a 6% reduction in flow in the southbound direction. Capacity values were provided by lane and they were compared to the Highway Capacity Manual (HCM, 2000), and the German Handbuch für die Bemessung von Strassenverkehrsanlagen (HBS, 2002). The capacity values for the median lane were consistent with both the HCM and the HBS values. The capacity value for the middle lane was consistent with the HBS but slightly lower than the HCM. The shoulder lane capacity was consistently lower than both manuals. It was concluded that there was no improvement in the capacity values over recognized standards.

In the Netherlands, a VSL system was installed along the A16 motorway near Breda in 1991. This system was designed to improve driving safety during fog conditions. The system has signs every 0.4-0.5 miles over 7.4 miles, 20 visibility sensors, and automatic incident detection. The speed limit was reduced to 80 kilometers per hour (km/h) (50 mph) from 100 km/h (62 mph) if visibility dropped below 140 meters, and was reduced to 60 km/h (37 mph) from 100 km/h (62 mph) if visibility dropped below 70 meters. When an incident was detected, a speed limit of 50 km/h (31 mph) was posted on the first sign upstream and 70 km/h (43 mph) on the second sign upstream (Robinson, 2000). The results of an evaluation (Zarean et. al, 1999) showed that drivers reduced their mean speeds by about 8-10 km/h (5-6 mph) during fog conditions. No information could be found on the current status of the system, but it was operational in 2000.

Another VSL system was installed in the Netherlands along a 20 km (12 mi) rural section of the A2 motorway between Amsterdam and Utrecht in 1992 (Robinson, 2002). The system is designed to reduce the risk of shockwaves, crashes, and congestion. Variable message signs are spaced approximately every one kilometer and loop detectors spaced every half kilometer. The

posted speed limits are determined by a system control algorithm based on 1-minute averages of speed and volume across all lanes. If an incident is detected, a speed of 50 km/h (31 mph) is displayed. The evaluation results showed that the severity of shockwaves and speed in all lanes were reduced (Van de Hoogen and Smulders, 1994). The vehicle speed and speed deviation decreased leading to fewer short headways as well as reduced severity of shockwaves. The study showed no positive effect on capacity or flow, but cited the safety benefits of traffic homogenization.

Speed limits were adjusted in England in response to the level of congestion on the M25 motorway in 1995. The objective of the system was to smooth traffic flow by reducing stop-start driving. The 22.6 km long system has VSL stations spaced at 1 km intervals, loop detectors at 500-meter intervals, and closed circuit television. Using loop detectors measuring traffic density and speed, speed limits are lowered in increments as congestion increases. The speed limits are lowered from 70 mph to 60 mph when volume exceeds 1,650 veh/h/ln, and lowered to 50 mph when volume exceeds 2,050 veh/h/ln. Results showed that traffic accidents decreased by 10-15% and there was a very high compliance with the VSL system (Robinson, 2000). The VSL system is still functioning today.

Rämä (1999) investigated the effects of weather-controlled speed limits and signs on driver behavior on the Finnish E18 site in Finland. The study looked at two scenarios compared to a control case: one in the summer where the maximum speed limit is 120 km/h (75 mph), and one in the winter where the maximum speed limit is 100 km/h (62mph). The control cases were during normal operating conditions in the summer and winter months. In the winter, during adverse road conditions the speed was lowered from 100 km/h (62 mph) to 80 km/h (50 mph). A 3.4 km/h (2.1 mph) decrease in speeds was observed. It was noted that during adverse conditions

that are harder to observe by drivers (such as “black ice”), the VSL was very effective at reducing speeds compared to the control case. It was concluded that the system is very beneficial for improving safety when drivers have a difficult time perceiving adverse conditions. In the summer, results showed that the 85th percentile speed was decreased more than the mean speed, essentially reducing high end speeds. Both winter and summer scenarios showed that VSLs decreased the mean speed and standard deviation of speeds and demonstrated traffic homogenization. This was an experimental site and no information could be found as to the current status of the system.

VSLs have been implemented in Sweden at 20 locations. Lind (2006) looked at the impacts of weather controlled VSLs on the E6 motorway in Halland, and the traffic controlled VSLs on the E6 in Mölndal, south of Gothenburg. The E6 in Mölndal is a low-speed urban motorway with normal speed limit of 70 km/h (43 mph). The VSLs in Mölndal were implemented as advisory speed limits in 2004 and changed to enforceable speed limits in 2006. This was part of a study to determine how VSLs were perceived by motorists in both enforceable and advisory conditions. The speed limit for free flow conditions was raised to 90 km/h (56 mph). In dense traffic the speed is reduced in a stepwise manner. At 950 veh/h/ln the speed is reduced to 70 km/h (43 mph) and can be reduced to 50 or 30 km/h (31 and 17 mph) depending on the density. Two thirds of interviewed drivers indicated that they supported the VSL system and said that it made them more attentive as to changes in traffic conditions. The same proportion reported a less hectic driving scenario and reduction of queue lengths. When the advisory speed limit was displayed crashes were reduced by 20% and when the enforceable speed limit was displayed crashes were reduced by 40%. The results showed an increase in average speed for all driving conditions and as much as a 40 km/h (25 mph) increase in potential

queue formation scenarios. The study concluded there was an improvement in driving behavior for congested conditions, and a homogenization of traffic.

Papageorgiou et al. (2008) studied the impact of VSLs on traffic flow behavior (flow-occupancy diagrams) through simulation of a motorway in Europe. The displayed speed was based on a threshold control algorithm, with possible speed limits of 60 mph, 50 mph, and 40 mph. The study showed that the 50 mph setting showed the most changes in traffic flow that could be used for improving traffic efficiency. The 40 mph setting was useful at high occupancies for displaying safe speeds, but not for improving traffic efficiency. The average occupancy was found to be higher when the VSL is implemented. The study concluded that the effect on capacity was not clear.

In summary, VSLs have been implemented in numerous areas throughout the United States, and are widespread throughout Europe. Most of the VSL systems in the US have been implemented to address adverse weather conditions. Several of the European systems however have been implemented to smooth flow and reduce congestion-related crashes. Several studies showed that mean speeds decrease when a VSL is implemented. Several studies showed the speed standard deviation to decrease as well, and that decrease has been associated with safety benefits. From the literature review it was not clear whether evaluation studies examined speeds upstream of the bottleneck and the impacts of the VSL both in space and time. Speed drop has typically been evaluated in terms of whether the speed limit was effectively reduced, but it is not clear whether the average speeds and/or travel times have been evaluated for the duration of the peak period and considering the entire section typically affected by congestion.

There has been little evidence to suggest that implementing VSLs has the potential to increase capacity. The systems using weather and road conditions to display VSLs have been

shown to reduce crashes and homogenize traffic conditions. Among active systems, the minimum speed limits provided in the US are typically between 40 mph and 50 mph, while those in Europe typically vary between 60 km/h (37 mph) and 80 km/h (50 mph). It is also common in European systems to display a speed of 50 km/h (31 mph) during a detected accident scenario.

2.2 VSL Simulation

Simulation is a valuable tool for assessing the impact of changes in the transportation system and selecting optimal alternatives without actually implementing and testing them in the field. Several studies have been conducted to evaluate various VSL algorithms prior to their implementation. This section provides an overview of such studies and summarizes their findings.

Hegyí et al. (2003) present a predictive model for coordination of VSLs to suppress shockwaves at highway bottlenecks. The objective of this control mechanism is to minimize the time a vehicle spends in the given network. The METANET model is used to simulate the network, but was modified to incorporate the effect of speed limits into the calculation logic. METANET is a second order macroscopic traffic flow model. The controller predicts the evolution of the network based on the current state of the network and a control input. The algorithm bases speed increments through real time calculations of traffic flow, density, and mean speed. Safety constraints are implemented into the model to prevent large speed limit fluctuations (e.g., 10 km/h). The model was applied to a benchmark freeway segment consisting of two nodes connecting one link. The study compared the use of continuous valued speed limits and discrete valued speed limits to a base scenario with no control. The results showed that in all control cases the coordination of speed limits eliminated the shockwave, and restored the volume exiting the section to capacity sooner.

Hegyi et al. (2005) continued work on model predictive control through coordination of VSLs and ramp metering. The study compared the results of simulated ramp metering, and ramp metering with VSLs on a simple network. The results showed that when used in conjunction the total time spent in the system was lower and resulted in higher outflow. The decision of which method to use depends on the demand of the on-ramp and the freeway. It is suggested that VSLs should be used if speed limits can limit the flow sufficiently, however if the flow becomes too large, ramp metering should be implemented. The authors suggest that integrated use of both technologies will produce more favorable results than the use of each technology by itself.

Lin et al. (2004) presented two online algorithms for VSL controls at highway work zones. The first VSL algorithm was aimed to reduce approaching traffic speed so as to increase the average headway for vehicles to merge onto adjacent lanes. It consisted of two modules: one to compute the initial speed of each VSL sign, and the second responsible for updating the displayed speed on each VSL sign. The algorithm computes the appropriate speeds starting on the link directly upstream of the work-zone. The algorithm computes the target density and appropriate speed for that segment and works upstream to calculate appropriate speed limits. The second VSL algorithm was aimed to maximize the total throughput from the work zone under some pre-defined safety constraints. The model looks at projected queue lengths and changes the upstream speed control signs based on the optimization of a throughput function. The simulation results by CORSIM indicated that VSL algorithms can increase work-zone throughputs and reduce total vehicle delays. Moreover, when VSL was implemented, speed variances were lower than other non-controlled scenarios, although the average speed didn't change significantly.

Lee et al. (2004) used a real-time crash prediction model integrated with the microscopic simulator PARAMICS to assess the safety effects of VSLs on a 2.5 km stretch of a sample

freeway segment. The algorithm for changing speeds was relatively simple. Three detector locations relay information to the controller which averages their values into one crash potential value. A crash threshold is predefined, and when the crash potential exceeds this threshold the speed limit for all three detector locations was set based on a set of criteria. When crash potential exceeded the threshold, the speed limits were reduced from the design speed limit (90 km/h) based on the average speeds: reduced to 50 km/h if average speed ≤ 60 km/h, reduced to 60 km/h if average speed > 60 and ≤ 70 km/h, reduced to 70 km/h if average speed > 70 and ≤ 80 km/h, and reduced to 80 km/h if average speed > 80 km/h. The results found that reduction in speed limits can reduce average total crash potential, and the greatest reduction in crash potential occurred at the location of high traffic turbulence such as a bottleneck. However, the reduction in speed limit also increased the travel time. Thus, there was a trade-off between safety benefits and system travel time increase. The results were not based on real traffic data and many assumptions in the simulation were not calibrated to field conditions. The authors speculated that this may account for the increase in travel time.

Lee et al. (2006) continued work using the simulator PARAMICS in combination with the real-time crash prediction model described earlier, to analyze the effect of VSLs on safety. Simulation results showed that the system obtained the greatest safety benefit when speed changes were gradually introduced (5 mph every 10 minutes). It was also found that it is best to base the displayed speed on the average speed of detectors immediately upstream and immediately downstream of the VSL location. However, the study has several limitations. First, it assumed that drivers would comply with the speed limit. Second, it ignored the potential of 'driver compensation' (driving faster downstream after reducing speed).

Mitra and Pant (2005) evaluated the impact of a VSL system on a freeway work zone using the model VISSIM. The authors considered three scenarios: base scenario (no work zone), reduced speed on the work zone link, and reduced speed with reduced lane width. The displayed speed was only changed through the work-zone and only one value indicating lowered speed was displayed. Through analysis of the data, a process was carried out for developing an equation to calculate expected delays for a reduced speed through a work zone. The authors concluded that this equation could help determine the proper speed through a work-zone without the use of repeated simulation.

Abdel-Aty et al. (2006) evaluated the safety effects of VSLs on I-4 in Orlando, Florida using PARAMICS. This was part of a series of papers which reported research related to the I-4 system. The algorithm not only investigated lowering speeds upstream of congestion, but also raising speeds limits after a congested area. The VSL signs were changed based on data from a detector directly associated with the sign. The study evaluated two speed regimes: low speed, and medium to high speed. The results found that there was a safety benefit in medium-to-high-speed regions but not in low-speed situations (congested situations). It was also shown that the greatest improvement in safety was achieved by abruptly changing speeds (15 mph) rather than gradually changing them. A travel time study was also conducted and showed a significant reduction in travel time through the segment. It was further recommended that decreasing speed limits before congestion and increasing them after congestion has positive impacts on safety and travel time.

In a subsequent study, Abdel-Aty et al. (2008) studied the effects of VSL on reducing crash risk on I-4 at different volume loading scenarios using PARAMICS. There were a total of 24 treatments in the experiment based on the extent of speed change, speed change distance, and

speed change duration (5 to 10 minutes). The study investigated the benefits of reducing the speed (5 -10 mph) entering a congested area and increasing the speed (5 mph) past the congested area. Crash risks were computed from a crash prediction model that was based on traffic parameters. The study found that VSLs could reduce the rear-end and lane-change crash risk at low volume conditions, especially when lowering the upstream speed limit by 5 mph and raising the downstream speed limit by 5 mph. Again, VSLs were not found to be effective in reducing crash risk during congested conditions.

Abdel Aty and Dhindsa (2007) also conducted a micro-simulation study using PARAMICS in order to determine the impact that VSLs and ramp metering would have on the safety of a 9-mile stretch of I-4 in Orlando. The study also investigated the impact of VSLs and ramp metering on operational parameters like speed and travel time. The speed limits were changed based on thresholds of 5 minute averages of travel speed, and the ALINEA feed-back algorithm was used for the ramp metering. It was concluded that implementation of VSL can increase average speeds and decrease speed variation in the network as well as improve the risk index. It was also shown that the best implementation strategy is one where the speeds are incremented by 5 mph over a half mile. It was also shown that for safety improvements, a scenario where only downstream speeds are increased, outperformed a scenario where upstream speeds are decreased and downstream speeds increased. A third conclusion drawn by the authors was that VSL and ramp metering are more effective when integrated together. When used in conjunction they showed shorter travel times and higher speeds than ramp metering or VSL alone.

Jiang and Wu (2006) used a cellular automaton model and showed that using multiple speed limits (where the speed limits decrease gradually from upstream to downstream) can help

remove traffic jams. For a single small jam the concept is that by altering the speeds appropriately one can decrease the inflow toward a jammed area and increase the outflow. This will eventually result in the jam being dissipated. Their model was not based on field data.

Allaby et al. (2007) evaluated the impact of a candidate VSL system on an 8-km section of the eastbound Queen Elizabeth Way, an urban freeway in Toronto, Canada. The study was conducted using the microscopic simulator PARAMICS combined with a categorical crash model developed by Lee (2003). The VSL algorithm used was based on a logic tree that uses threshold values for flow, occupancy, and average travel speed. The base speed used was 100 km/h (62 mph) and it could be reduced to 80 km/h (50 mph) and 60 km/h (37 mph). The signs were arranged so there was never an abrupt change of speed limits (10 km/h difference) between signs. Each VSL sign was linked to an adjacent loop detector, and each sign operates individually. The results of the simulation showed that implementation of VSL signs could significantly improve safety, however the authors concluded that the use of VSL signs increased the travel time for all traffic scenarios considered.

Piao and McDonald (2008) assessed the safety benefits of in-vehicle VSLs on motorways using the microscopic simulation model AIMSUN. Traffic on UK motorway M6 with speed limit of 70 mph was simulated under different scenarios. VSLs were applied when the speed difference between a queuing section and the upstream section was larger than 20 km/h (12.4 mph), and were provided to drivers through in-vehicle information. The simulation assumed that all vehicles were equipped with the in-vehicle devices. The adjusted speed limits could be 60 km/h (37 mph), 70 km/h (43 mph), 80 km/h (50 mph), 90 km/h (56 mph), or 100 km/h (62 mph). The simulation results showed that VSL reduced speed differences creating homogenization, reduced very small time headways, small time-to-collision (TTC) events, and lane change

frequency. This in effect reduced crash potential. The authors also indicated that there were potential safety risks in using the in-vehicle VSL compared with roadside VSL: large speed variations in speed could occur because some vehicles didn't have the in-vehicle device.

Papageorgiou et al. (2008) used a quantitative model to investigate the impact of VSL implementation on traffic flow. VSLs were incorporated into the general second-order traffic flow model METANET as a control component. The study evaluated the system based on a no-control case, coordinated ramp metering, VSL, and integrated scenario. The freeway was set up as a constrained discrete-time optimal control problem and solved using a feasible direction algorithm. It was shown that VSLs can substantially improve the traffic flow efficiency of a stretch of roadway especially when combined with coordinated ramp metering. The study concluded that when the optimal solution is applied to real motorway traffic, the solution will inevitably become non-optimum due to uncertainties in the real traffic stream. The authors suggested that future research could use the optimal solution to develop a suitable feedback control strategy and update the solution in real time.

Carlson et al. (2010) expanded on the work of Papageorgiou et al. (2008) by using a similar method, to explore the parallels between ramp metering and applying VSL upstream of a potential bottleneck or high volume merging situation. The METANET second order macroscopic model was altered to allow the VSLs to be incorporated. The study showed that when applied upstream, the VSL can act similarly to ramp metering where the flow is held back on the mainstream rather than on the ramp. The traffic arriving at the bottleneck is temporarily reduced and the system delays propagation of the congestion. Four scenarios were evaluated: no-control, VSL control, ramp metering, and integrated control. The VSL case decreased total time spent in the system (TTS) by 15.3%, and when VSLs and ramp metering are used in conjunction

the TTS was reduced by as much as 19.5%. The study concluded that traffic flow and capacity can be improved through VSL use by reducing the capacity drop at bottlenecks. However, if the VSL is applied at under-critical conditions without the potential for bottleneck mitigation, mean speed is lowered and flow efficiency is decreased.

Popov et al. (2008) proposed a speed limit control approach to eliminate shockwaves based on a distributed controller design. The METANET environment was used for the simulation. In this design, each VSL sign has its own controller, but they all use the same structure and parameters. The proposed method requires using the appropriate amount of upstream and downstream data. Different scenarios were presented where each controller uses data from as many as 5 downstream controllers and one upstream controller. The maximum speed limit was 120 km/h (75 mph), and could be lowered in increments of 10 km/h to a minimum of 50 km/h (31 mph). The authors showed that a simple, linear, static controller using immediate neighbor information successfully resolves a shockwave. The control scenario when compared to a scenario without controllers reduced total time spent in the network by 20%.

Ghods et al. (2009) used METANET to investigate the use of ramp metering and VSL in order to reduce peak hour congestion. An adaptive genetic-fuzzy control was used and was compared to the traditional ALINEA controller. Local density, local speeds, and queue length of the on-ramp were used as input data to develop the fuzzy controller. The fuzzy controller processes this input data and provides a corresponding metering rate and two VSLs. The idea behind fuzzy logic is to have a controller that resembles human decision making. It can process imprecise input data to arrive at a definitive conclusion. Rather than having precise threshold values that determine the output values of the controller, approximate multi-valued boundaries are used. This allows for input data to have partial membership to a category as opposed to the

traditional “crisp” membership or non-membership options only. The study showed that the genetic fuzzy ramp metering and VSL control improved TTS by 15.3%.

In summary, much research has been conducted on the potential benefits of VSLs through the use of simulation. One set of studies has used VSLs as a control mechanism similar to that employed in ramp metering. These studies concluded that VSLs can be used to suppress shockwaves at bottlenecks by implementing the VSL upstream of a bottleneck. Those studies reported that VSLs were effective in reducing TTS in the network, and their effect was more beneficial when combined with ramp metering. Another set of studies investigated the use of VSLs in micro-simulators (VISSIM, PARAMICS, AIMSUN) and evaluated the safety benefits of such systems. These studies generally concluded that VSLs can improve safety, as they tend to reduce speed variability.

2.3 VSL Algorithms

This section provides more detailed information regarding various VSL algorithms that have been developed. Different algorithms have been developed based on the purpose of the VSL. The first part of this section discusses VSL algorithms developed to mitigate congestion and improve safety, while the second part focuses on algorithms developed to address weather and other issues.

2.3.1 Congestion and Safety-Related Algorithms

The following three algorithms aim to mitigate shockwaves and are based on a combination of parameters:

- Along A2 between Amsterdam and Utrechtin, 1992 Netherlands (implemented)
 - Based on 1-minute averages of speed and volume across all lanes
 - 50 km/h if incident occurs
 - Severity of shockwaves and speed in all lanes were reduced
 - Detailed information regarding location of signs and detectors was not provided

- METANET simulation 2003 (not implemented)
 - Bases speed increments through real time calculations of traffic flow, density, and mean speed
 - Uses rolling horizon values to continuously update the optimal solution
 - Showed that during a developing shockwave the model predictive control created a scenario with less congestion and higher outflow
- METANET simulation 2008 (not implemented)
 - Used individual controller for each VSL sign using data from as many as 5 downstream controllers and one upstream controller
 - Reduced speeds in 10 km/h increments from 120 km/h to as low as 50 km/h
 - Showed that a simple, linear, static controller using immediate neighbor information successfully eliminates a shockwave

The following two algorithms are based on flow:

- M25 Motorway, 1995 England (implemented)
 - When flow > 1650 veh/h/ln: 70 mph to 60 mph.
 - When flow > 2050 veh/h/ln: lowered to 50 mph
 - Accidents decreased by 10-15%, very high compliance
 - Detailed information on location of signs and detectors not provided
- On the E6 motorway in Mölndal, 2006 Sweden (implemented)
 - Free flow = 90 km/h
 - 950 veh/h/ln = 70 km/h
 - Speed can be reduced as low as 50 to 30 km/h
 - When speeds were advisory there was a 20% crash reduction observed. For enforceable speed limits the crash reduction improved to 40%. Other impacts included average speed increase, homogenization of traffic, and reduction in queue length.

The following algorithm is based on average occupancy thresholds:

- I-4 in Orlando, FL 2009 (implemented)
 - Classifies traffic as either free, light or heavy depending on occupancy value
 - <16% = free
 - >16% and < 28% = light
 - >28% = heavy

- Reduces speeds in 10 mph increments from 50 mph to 30 mph

The following algorithm is based on average travel speeds:

- PARAMICS simulation 2004 (not implemented)
 - Each VSL has an associated loop detector located adjacent to it
 - Three signs are grouped together and data for these signs was averaged into one value
 - If a crash potential threshold is reached the displayed speed is dropped at all signs using a set of criteria (all signs display the same speed)
 - 50 km/h if avg. speed \leq 60 km/h,
 - 60 km/h if $60 <$ avg. speed \leq 70 km/h,
 - 70 km/h if $70 <$ avg. speed \leq 80 km/h,
 - 80 km/h if avg. speed $>$ 80 km/h
 - Reduced average total crash potential, especially at the bottleneck, but increased the overall travel time

This algorithm is based on a combination of flow, occupancy, and average speed, using a logic tree.

- PARAMICS simulation 2007 (not implemented)
 - Each VSL sign is linked to an adjacent detector that operates individually
 - For low volumes (less than 1,600 vehicles per hour per lane (vphpl)) occupancy is used as part of the criterion for reducing speeds. For higher volumes (more than 1,600 vphpl) occupancy is not considered.
 - Ultimately average speed determines the displayed speed. This algorithm does not address gradual speed limit reduction as drivers are approaching the bottleneck.
 - The simulation results showed that VSL signs could improve safety but that the travel time for all traffic scenarios considered were increased.

2.3.2 Weather-Related and Other Algorithms

The following four algorithms were developed to address weather-related issues (visibility, wind speed, precipitation severity, etc.):

- Along A16 motorway near Breda, 1991 Netherland (implemented)
 - 100 km/h (normal)
 - 80 km/h if visibility <140 meters
 - 60 km/h if visibility < 70 meters
 - Mean speeds reduced by about 8-10 km/h during fog conditions
- 25 km, between Hammina and Kotka, 1997 Finland (implemented)
 - 120 km/h for good road conditions
 - 100 km/h for moderate road conditions
 - 80 km/h for poor road conditions

On the Finnish E18 site, 1998 Finland (implemented)

- Lowered from 100 to 80 km/h in winter
- Lowered from 120 to 100 km/h in summer
- Decreased both the mean speed and the standard deviation of speed

Along a 19-mile section of I-75, 1993 Tennessee (implemented)

- 5 to 10 percent reduction in speed
- no crashes due to fog after implementation

In summary, there are a number of existing algorithms based on different performance measures. For algorithms involving congestion mitigation or shockwave dampening, VSL signs are almost always associated with downstream detectors to decrease flow entering a congested area. Algorithms based on weather or road condition parameters usually deal with VSLs associated with adjacent detectors. In both cases it is most common to gradually lower the speed limit in increments of 5 or 10 mph. Most algorithms also use a safety measure that prevents adjacent signs from having more than a 10 mph difference between them. In addition, nearly all systems use a mechanism to prevent hysteresis, or rapid fluctuation between displayed speeds. Some systems use minimum time durations, and others use reverse thresholds to avoid this event.

2.4 CORSIM Simulation Software

CORSIM is a microscopic simulation program used to simulate a variety of traffic situations. CORSIM's RTE allows the user to interface directly with the CORSIM simulation tool and implement a variety of algorithms that can bypass CORSIM standard algorithms. RTEs

have been used to run simulations with actual hardware in the loop, and to simulate other ITS tools that are not part of CORSIM's user interface. The way in which an RTE works within the context of the TSIS shell is displayed in Figure 2-2 (McTrans, 2009).

The TShell is where users can interface with the program and specify inputs to the simulation. It sends the input data to the CORSIM Driver Component which coordinates with the CORSIM property pages. The driver communicates with the CORSIM server that calls a series of exported functions that drive the simulation loop. At the same time the server calls the specified RTE functions. The CORSIM simulation has a number of specified call points where it can export RTE functions. These points include an initialization at the beginning of the simulation, after the completion of a time step, or at the completion of the simulation. There is a complete list of the call points in the CORSIM RTE developer's guide. The CORWin interface allows the RTE to send messages to the CORSIM driver, which can display these messages on the end user screen. Data structures can be accessed by the RTE through shared memory. This gives access to and allows the RTE to control aspects of the simulation. RTEs are a powerful tool within CORSIM that allow the user to simulate new technologies and ideas (McTrans, 2009).

2.5 Literature Review Summary

It is clear VSLs are prevalent in Europe and are becoming more popular in the United States. It is also clear that simulation is a powerful tool for assessing new technologies that have the potential to improve traffic flow and safety. With this growing interest in VSLs, the need to assess their benefits with simulation becomes evident. Creating a framework to simulate VSLs in CORSIM will provide a convenient way to test various algorithms and assess potential benefits before actually implementing these strategies on a roadway.

Table 2-1. Orlando, Florida I-4 VSL control thresholds

	Occupancy for decreasing speed limit (%)	Occupancy for increasing speed limit (%)	Speed limit (mph)
Free flow	< 16	< 12	50
Light congestion	16 - 28	12 - 25	40
Heavy congestion	> 28	>25	30

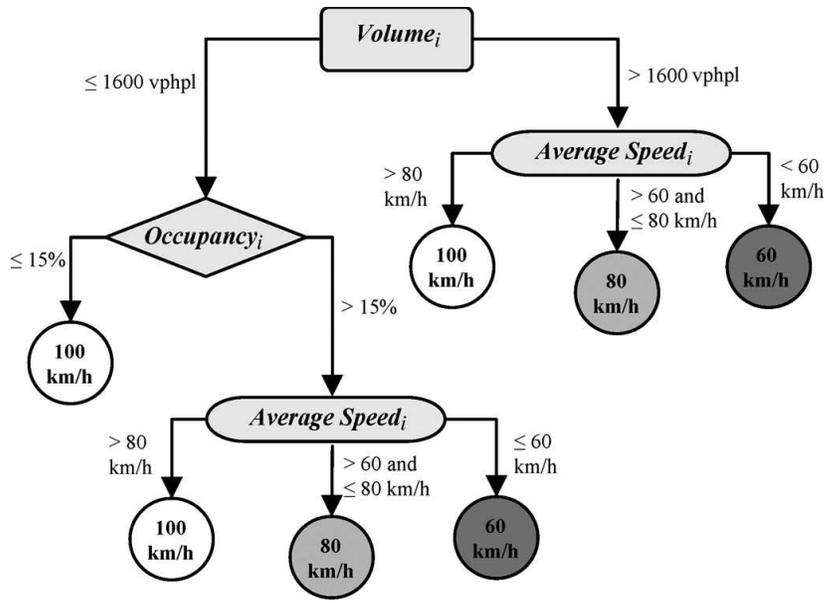


Figure 2-1. Decision path for determining the new posted speed of the trigger VSLs. [Adapted from Allaby, P., Hellinga, B., and Bullock, M., 2007. Variable Speed Limits: Safety and Operational Impacts of a Candidate Control Strategy for Freeway Applications. IEEE Transactions on Intelligent Transportation System 8 (4), pp.671-680, Figure 3.]

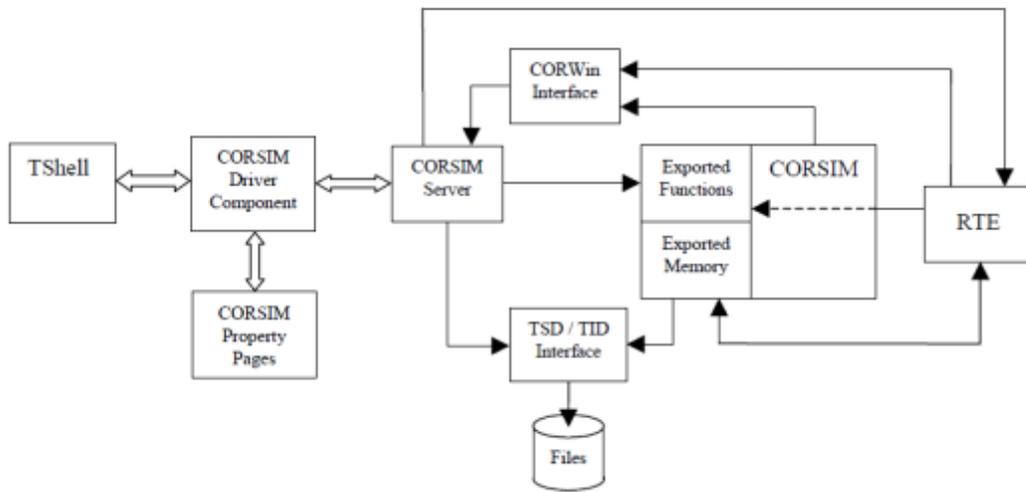


Figure 2-2. Operation of CORSIM and RTEs [Adapted from The McTrans Center., 2009. Run-Time Extension User's Guide. University of Florida, Gainesville, Florida. pp. 2-3, Figure 1.]

CHAPTER 3 METHODOLOGY

The research was executed in the following four steps: study site selection and data assembly, development of the run time extension (RTE), implementation/testing of variable speed limit (VSL) algorithms, and analysis of simulation results. Each of these steps is described briefly in the following paragraphs.

3.1 Study Site Selection and Data Assembly

A 13-mile section of Interstate 95 in Miami, Florida was selected as the study site. This study site was selected based on the availability of data, the existence of recurring congestion along the site, and the interest of FDOT to evaluate whether VSL would be an effective strategy at this location. The site experiences recurrent congestion at several locations along its length. A calibrated Corridor Simulation (CORSIM) simulation network of this stretch of roadway was available, and was used as the basis for simulating the selected VSL algorithms. Detailed information regarding the study site is provided in Chapter 4 of the thesis.

3.2 Development of RTE

A total of three algorithms, which vary mainly in the type of input parameters used in decision making, were tested. From the literature reviewed it is desirable to have algorithms based on volume, average travel speed, occupancy, or a combination of the three. The first algorithm tested is based on occupancy thresholds. The second algorithm is based on volumes, and the third uses average travel speed as one of several parameters. Different threshold values were tested for each algorithm to determine the sensitivity of each algorithm to these thresholds. Variables associated with the sign spacing such as spacing and number of signs were also tested.

After the algorithms were selected and the specific scenarios finalized, the RTE module was built. This involves writing C++ code for the RTE to interact with CORSIM. There are

separate RTEs for each VSL algorithm and each scenario. The RTE uses inputs from detectors within the CORSIM simulation at every time step, and determines whether a speed change needs to be implemented. The RTE then outputs to the CORSIM simulation the updated speed limit for each VSL sign. Chapter 5 describes the RTEs developed in greater detail.

3.3 Testing and Analysis of Algorithms in CORSIM

The three identified algorithms were implemented into the CORSIM simulation through the RTEs developed. Specific scenarios were created for each algorithm to test the threshold values of each algorithm as well as the location of each detector and VSL sign. Each algorithm was run a sufficient number of times for each of the scenarios tested.

Output processing was performed for each run to generate outputs in the form of comma separated files. Contained in these files are traffic volumes, speeds, travel times, and miles traveled. The data were aggregated for each detector location and averaged over the total number of runs.

The results from the output processing were used to analyze the performance of each algorithm. Each scenario within a given algorithm is compared to the other scenarios of that algorithm. The scenarios are compared based on average travel speed, total travel time, vehicle miles traveled, and throughput. They are compared based on link performance measures surrounding the VSL implementation as well as overall system values to obtain the optimum threshold values for the algorithm. The scenario with the best network performance was selected and compared to the other optimized algorithms. Qualitative comparisons are also provided to assess the performance of each algorithm. Chapter 6 provides the results of the simulation and the comparisons outlined above, along with conclusions regarding the effectiveness of VSL and specific algorithms.

CHAPTER 4 STUDY SITE DESCRIPTION

This chapter first describes the study site and then provides an overview of the Corridor Simulation (CORSIM) network used in the simulation.

4.1 Study Site Selection

The selected study site is a 13-mile section of I-95 running through Miami, Florida from I-395 to Miami Gardens Dr. in the northbound direction. An aerial view of the section is shown in Figure 4-1. This section of I-95 has two High Occupancy Tolling (HOT) lanes, as well as ramp metering. The variable speed limit (VSL) control is limited to the general purpose lanes and does not have a direct effect on the HOT lane operations. The analysis of results is confined to the general purpose lanes. The Florida Department of Transportation (FDOT) is currently considering the implementation of VSLs along I-95 at this location. The roadway is already equipped with inductive loop detectors that can obtain speed, volume, and occupancy. The inductive loop detectors have ID numbers based on data from the Statewide Transportation Engineering Warehouse for Archived Regional Data (STEWARD) database. The locations of the loop detectors along the study section are shown in Figure 4-2.

The speed profile over the entire I-95 section being analyzed is displayed in Figure 4-3. These speeds are averaged over a 15-minute period during the onset of congestion. In the simulation this time period corresponds to time period 4. As shown there are two noticeable bottlenecks; one is located immediately before the exit to the turnpike, and the other is at the entry to NW 103rd street. The focus of this study is the bottleneck area just before the turnpike and the upstream area affected by it.

4.2 Calibrated CORSIM Simulation

A CORSIM model has already been developed to replicate current traffic operations on the stretch of I-95 (FDOT, 2011). A screen shot of the network is shown in Figure 4-4. The simulation replicates the entire 13-mile stretch. However, the VSL implementation is only implemented at one bottleneck location. The data used in the previously referenced study for calibration of the network was obtained from the STEWARD database. The data represents traffic conditions on October 7, 2009, from 3:30 p.m. to 6:30 p.m. Fifteen minute averages of these parameters were used to generate volumes input into the software. The data consisted of speed averages and volumes used to calibrate each 15-minute period. These three hours were selected to include the p.m. peak period and the associated congestion formation and dissipation. The network was calibrated to match field-recorded volumes and speeds over each time period, and it replicates both the ramp metering and HOT lane operations. The ramp metering uses a constant metering rate that is not demand sensitive, and thus it is not expected to interact with the VSL algorithms in these simulations. The HOT lanes are modeled as a separate parallel facility with interchanges at various access points. Calibration results for each time period are provided in Appendix A.



Figure 4-1. Section of I-95 being simulated by CORSIM

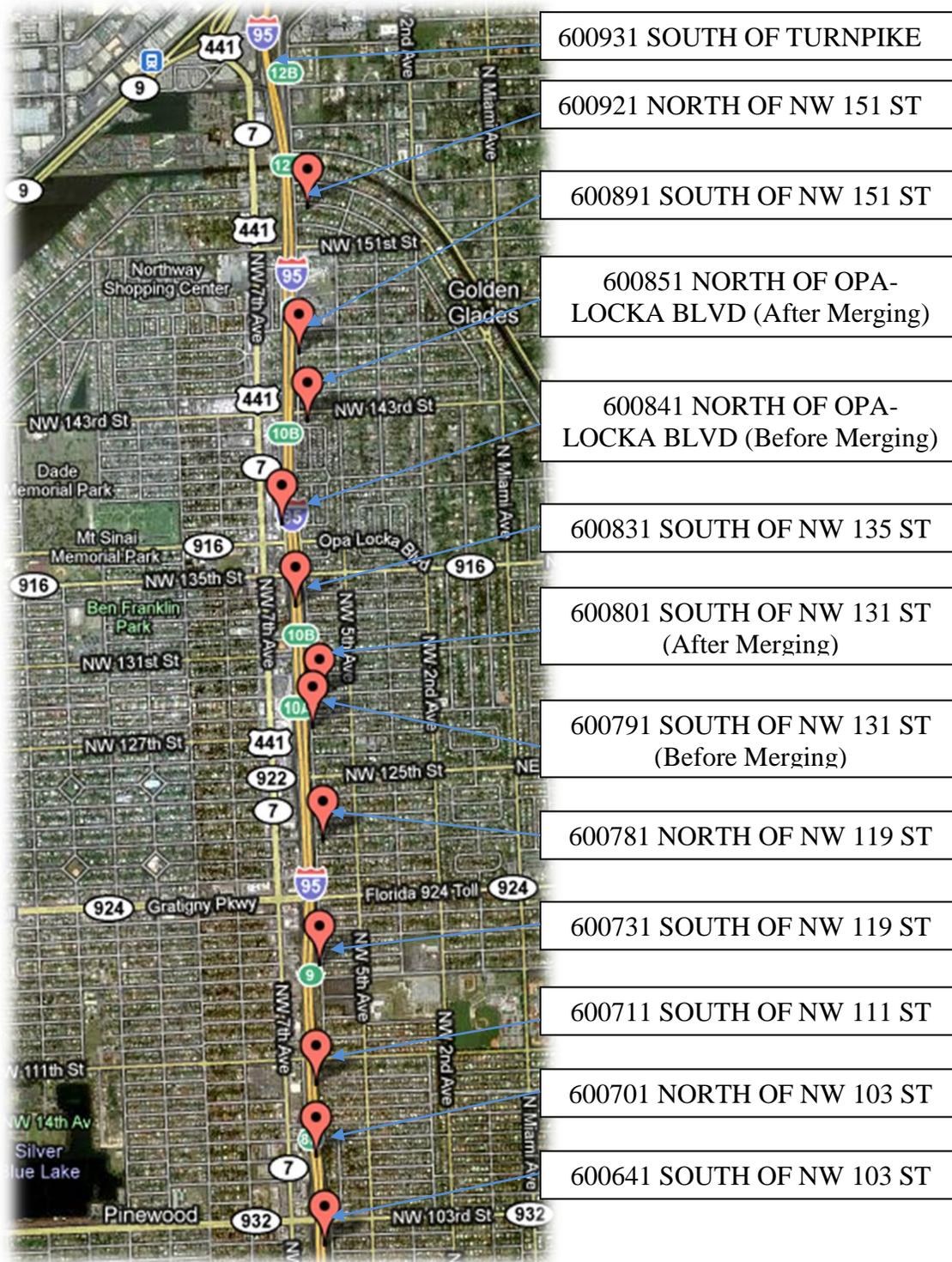


Figure 4-2. Location of detectors along the section of I-95 study site containing proposed VSL implementation

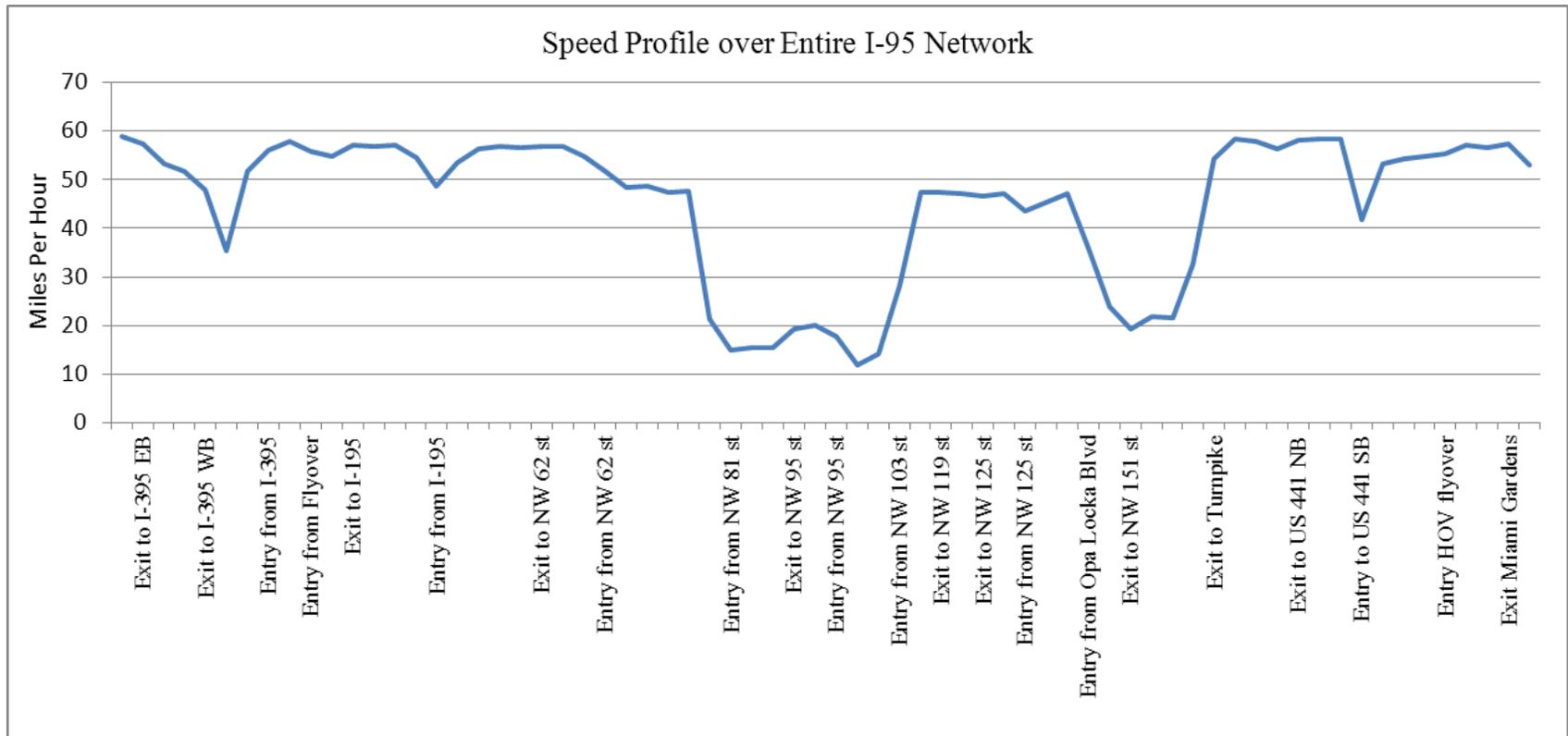


Figure 4-3. Speed profile of the I-95 section for time period 4

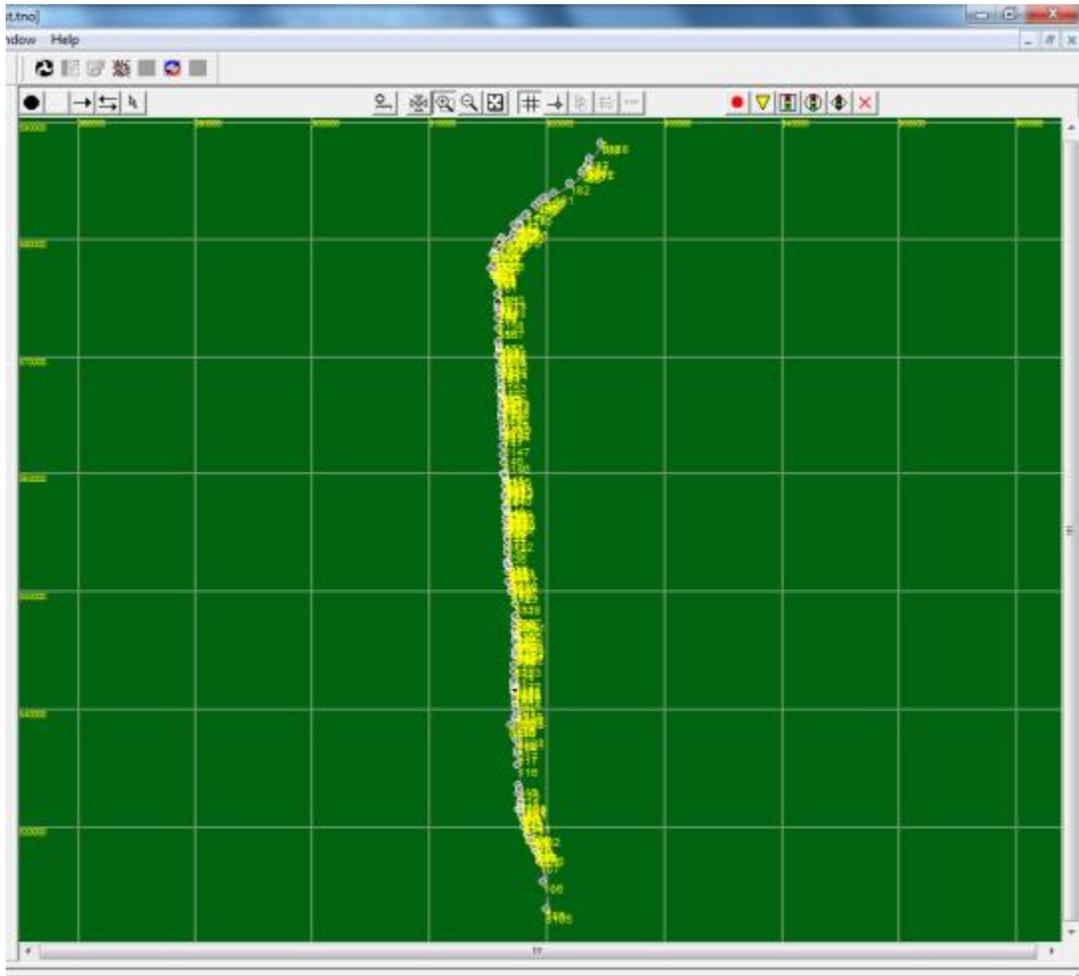


Figure 4-4. Screen shot of I-95 network represented in CORSIM

CHAPTER 5 DEVELOPMENT OF THE RTE

This chapter provides a description of the run time extension (RTE) development. First, the variable speed limit (VSL) algorithms used in the simulation are identified. Then the options for sign location variations tested with each algorithm are outlined. The construction and framework of the RTE programs, as well as an overview of how the programs are running are discussed in the last section.

5.1 Selection of Algorithms

Based on the literature review, a total of three algorithms are selected, which represent the major types of algorithms that have been tested and implemented elsewhere. The study selected algorithms that use different measures for triggering a speed limit change, to evaluate the impacts of different types of algorithms. The occupancy-based algorithm was selected as it is the one that is currently in use in Orlando, Florida (PBS&J, 2009). The volume-based algorithm was selected because it is implemented on the M25 in England (Robinson, 2000) with very good overall results. The third algorithm selected is based on a combination of flow, occupancy, and average travel speed, and it is based on a study of a freeway in Toronto, Canada (Allaby et al., 2007). This algorithm was selected because it seemed a promising alternative to the other two; however this one has not been implemented in the field.

Each algorithm has a range of threshold values that were tested. To prevent a rapid fluctuation of speed limits, each algorithm has one set of thresholds for lowering the speed limit and another set of thresholds for raising them. Each algorithm functions similarly within the freeway system. An inductive loop detector is located at the bottleneck, and relays 1-minute averages of speed, occupancy, and volume to a VSL sign upstream of this location. When a particular threshold value is reached the speed limit is reduced at the associated VSL sign.

Similarly, when a parameter drops below one of the reverse thresholds the speed limit is allowed to increase back to a higher speed. The speed limit is only allowed to drop by one increment at a time. For instance, if the current speed limit is 60 mph, and a threshold is reached that notifies the sign to drop to 40 mph, the speed is only reduced to 50 mph. This prevents drastic changes in the speed limit that might create driver confusion.

5.1.1 Algorithm Based on Occupancy

The algorithm based on occupancy has two sets of threshold values; one for the decreasing of speed limits and one for the increasing of speed limits. The VSL sign is linked to downstream detector location 600931, and the average occupancy is calculated over all of the lanes. The traffic is classified as either free-flow, light congestion, or heavy congestion. If the occupancy crosses a threshold line the speed limit is decreased by an increment of five miles per hour. Similarly the speed limit may increase back to its previous value but not more than 5 mph at a time. This algorithm is based on the current operating algorithm of the I-4 system (PBS&J, 2009). The I-4 implementation evaluates the speed limit every 120 seconds, and this study evaluates the speed limit every 60 seconds. The first threshold scenario uses the same values as the I-4 system. The next two threshold scenarios are generated based on findings from NCHRP Report 3-87 (Elefteriadou et al. 2009). That report reported occupancy thresholds as a function of the probability of breakdown at merge junctions. The three threshold scenarios are shown in Tables 5-1, 5-2, and 5-3.

5.1.2 Algorithm Based on Flow

The algorithm based on volumes also uses two threshold values; one for the decreasing of speed limits and one for the increasing of speed limits. The VSL sign is linked to downstream detector location 600931, and average volume is computed in vehicles per hour per lane. When a volume drops below a specified threshold, the speed limit is decreased according to the threshold

crossed. To return to the original speed the volume must cross a different threshold. This prevents a rapid fluctuation of speed limits without using minimum time duration. The first set of threshold values were obtained from a study conducted on the M25 in England (Robinson, 2000), and are shown in Table 5-4. The second set of thresholds are obtained from speed flow diagrams in the 2000 Highway Capacity Manual (HCM, 2000), and are shown in Table 5-5. The thresholds are obtained by locating the volume of traffic where speeds drop for a given free flow speed, using the associated volume as the threshold point.

5.1.3 Algorithm Based on a Logic Tree including Flow, Occupancy, and Average Speed

In this algorithm speed limits are determined based on a logic tree that includes flow, occupancy, and average travel speed. The decision making logic is shown in Figure 5-1. The algorithm first takes into account flow data from loop detector location 600921. If the volume is less than or equal to 1750 vphpl, the next step is to consider occupancy. If occupancy is less than or equal to 16%, the maximum speed limit is posted. If the occupancy is greater than 16%, average speed determines which speed is displayed. Going back to the first step, if the volume is greater than 1750 vphpl, the logic skips straight to the average speed calculation. The speed to be displayed is then sent to the appropriate VSL sign.

The thresholds shown in the figure are placeholders. The actual thresholds for volume and occupancy are the thresholds that display the best performance in the first two algorithms. The average travel speed thresholds tested are 50 mph, and 45 mph. This algorithm is based on research conducted on a candidate VSL system in Toronto, Canada (Allaby, 2007)

5.2 Sign Location Variations

Along with the different algorithms and thresholds tested, four different sign locations are also tested. Each scenario uses the same detector placement but varies in the location of VSL sign. The first scenario uses one VSL sign spaced approximately one-half mile from the

bottleneck source. The second scenario uses one sign spaced approximately one mile from the bottleneck source. The third scenario uses two signs, with the first sign placed approximately one-half mile from the bottleneck source. The second sign is placed approximately one-half mile upstream from the first sign location. The fourth scenario uses two signs, with the first sign placed approximately one mile from the bottleneck source. The second sign is placed approximately one mile upstream of the first sign. In the case of two signs operating, the upstream sign always displays a speed limit 5 mph higher than the downstream sign. For example if the downstream sign is displaying 55 mph, the upstream sign also displays 55 mph. If the downstream sign displays 50 mph, the upstream sign still displays 55 mph. If the downstream sign displays 45 mph, the upstream sign displays 50 mph. The four sign location scenarios are shown visually in Figures 5-2 through 5-5

5.3 Building of DLL's to communicate with the RTE interface

Implementing the algorithms on the Corridor Simulation (CORSIM) network requires a dynamic link library (DLL) that interfaces with the CORSIM simulation in real-time. CORSIM allows this DLL to be imported through an RTE interface. The interface allows the DLL to import and export variables internal to CORSIM.

Three different DLLs are built, one for each type of algorithm. The general structure of the program works similarly for each case, but the rules and thresholds for the speed change logic differ between each program. A flowchart displaying the general logic of the program is displayed in Figure 5-6.

Upon initialization of the simulation, the DLL program identifies where VSL signs have been specified, and what detectors are used to control the VSL operation. The links affected by the VSL sign are also identified. This allows the speeds on the downstream links to be updated when a speed limit change occurs. During the initialization period the point processing interval is

also defined. This determines how data is aggregated from the inductive loop detectors. For this set of scenarios the point processing interval has been set to 60 seconds.

After initialization is complete the DLL is accessed at the call point PREFRESIMVEHICLE. This is every time-step (one second) during the simulation before vehicle movement takes place. First the program checks to determine whether the simulation is still in the initialization period. If so, the program exits the function and this is reassessed at the next time step. If the simulation is not in the initialization period the current speed limit is assessed based on average values relayed from the specified inductive loop detectors. If it is determined that a speed change is to occur, the free-flow speed is updated on the link containing the VSL sign. At the same time a message is displayed as the simulation runs to indicate a speed change has occurred. An example of a speed change message during the CORSIM simulation with the occupancy algorithm is displayed in Figure 5-7.

After the free flow speed has been updated on the link containing the VSL sign, the free flow speed is updated on the downstream links. The free flow speeds at the downstream links are updated every 15 seconds, and the time of the speed change depends on the free flow speed and distance between the VSL sign and the downstream link. This creates a rolling speed change so that all the downstream links are not updated simultaneously. This method mimics a real world scenario where the first vehicle observing a speed change represents a rolling speed change through the downstream links. Figure 5-9 shows how the speed change would propagate downstream for a sample speed limit change. In the diagram the speed limit drops from 55 to 50 mph starting at the VSL sign. The downstream link speeds are then updated every 15 seconds based on the length of the link and the free flow speed.

Table 5-1. Occupancy thresholds for displayed speed limits (scenario 1)

Traffic Category	Occupancy for decreasing speed limit (%)	Occupancy for increasing speed limit (%)	Speed limit (mph)
Free flow	< 16	< 12	55
Light congestion	16 - 28	12 - 25	50
Heavy congestion	> 28	>25	45

Table 5-2. Occupancy thresholds for displayed speed limits (scenario 2)

Traffic category	Occupancy for decreasing speed limit (%)	Occupancy for increasing speed limit (%)	Speed limit (mph)
Free flow	< 10	< 8	55
Light congestion	10 - 30	8 - 27	50
Heavy congestion	> 30	>2	45

Table 5-3. Occupancy thresholds for displayed speed limits (scenario 3)

Traffic category	Occupancy for decreasing speed limit (%)	Occupancy for increasing speed limit (%)	Speed limit (mph)
Free flow	< 20%	< 17%	55
Light congestion	20 - 35%	17 - 32%	50
Heavy congestion	> 35%	>32%	45

Table 5-4. Volume thresholds for displayed speed limits (scenario 1)

Flow for decreasing speed limit (vphpl)	Flow for increasing speed limit (vphpl)	Speed limit (mph)
< 1650	-	55
> 1650	< 1450	50
> 2050	<1850	45

Table 5-5. Volume thresholds for displayed speed limits (scenario 2)

Flow for decreasing speed limit (vphpl)	Flow for increasing speed limit (vphpl)	Speed limit (mph)
< 1450	-	55
> 1450	< 1250	50
> 2000	<1800	45

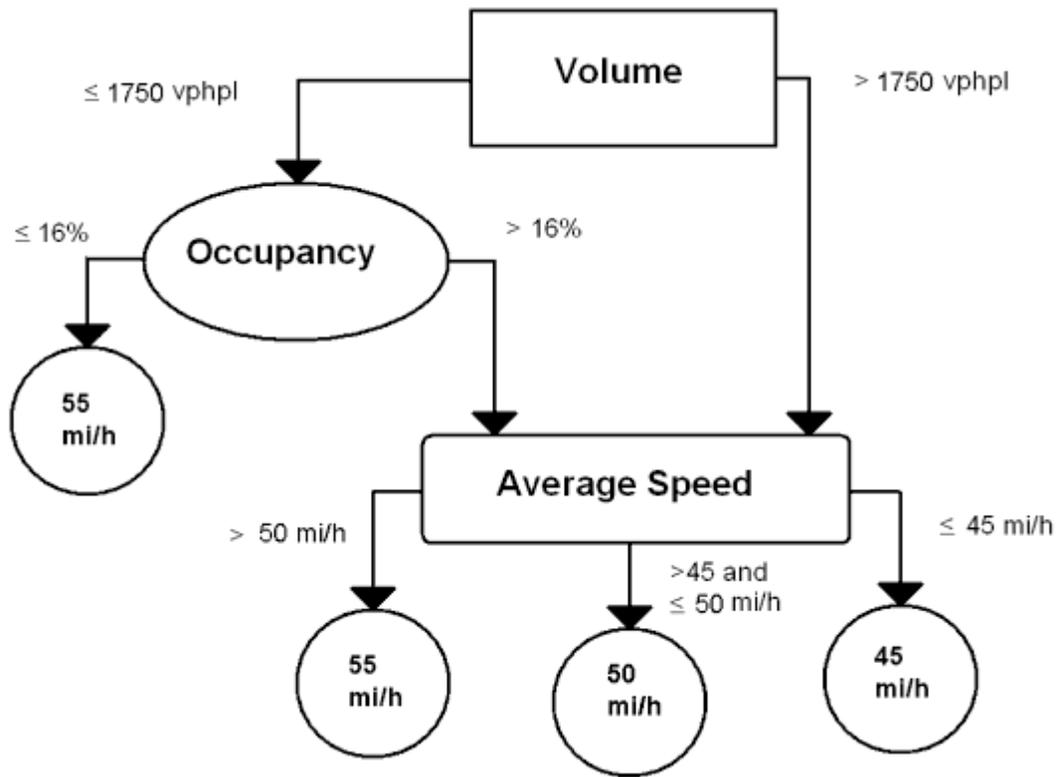


Figure 5-1. Decision tree logic for third algorithm

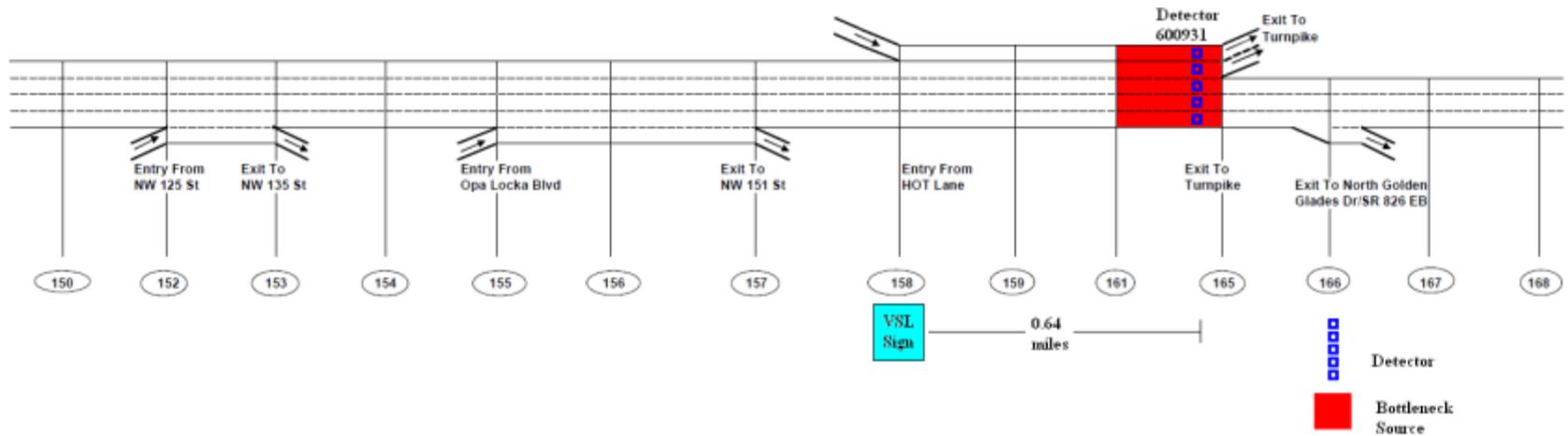


Figure 5-2. Use of one sign spaced approximately one-half mile

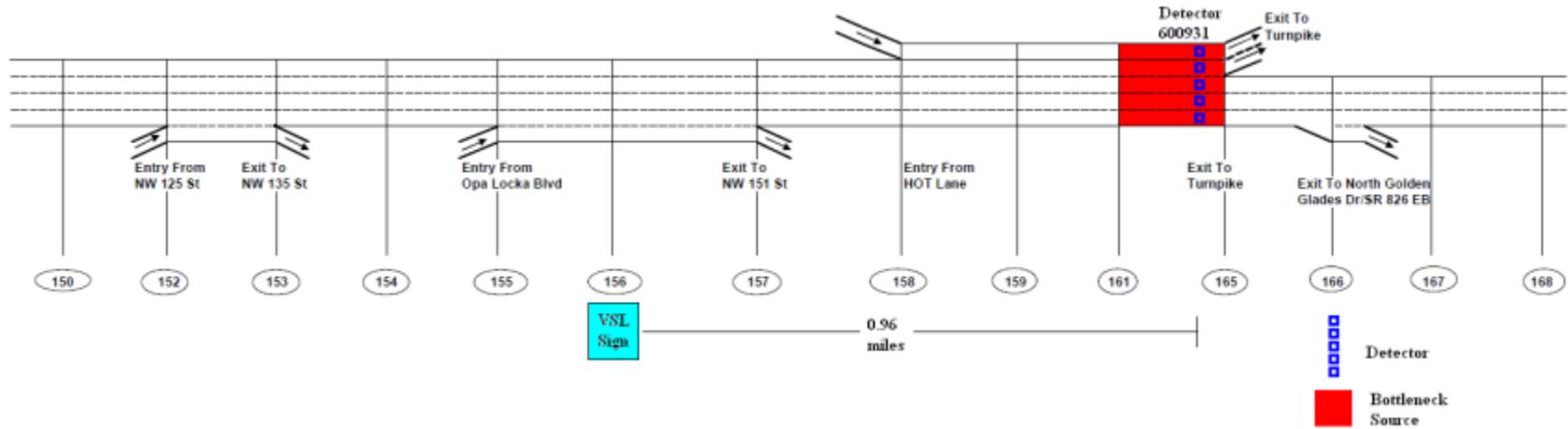


Figure 5-3. Use of one sign spaced approximately one mile.

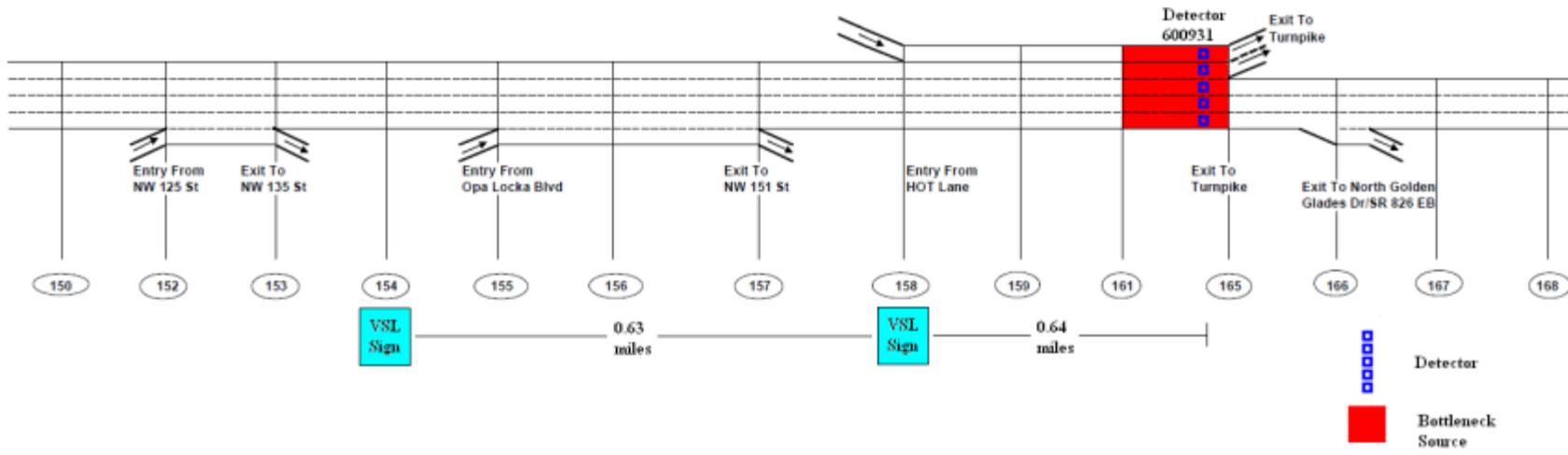


Figure 5-4. Use of two signs spaced approximately one-half mile apart

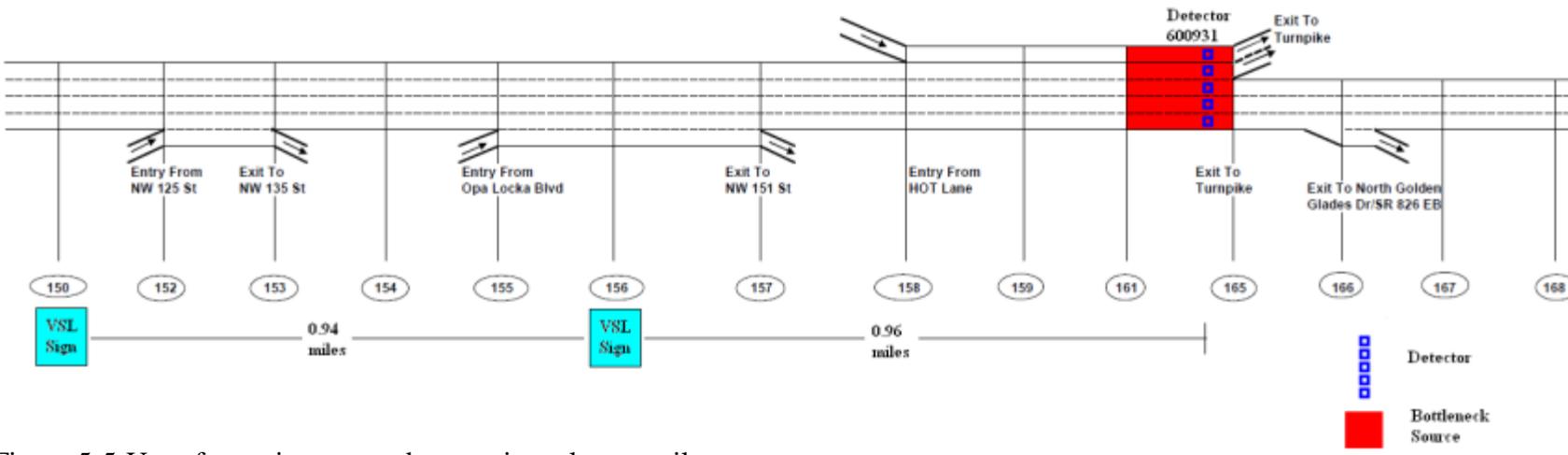


Figure 5-5. Use of two signs spaced approximately one mile apart

```

NETWORK HAS REACHED EQUILIBRIUM.
COMPLETED 60 SECONDS OF SIMULATION..... (CORSIM) ( 2.83 CPU Seconds)
Speed Limit has been reduced to 50 MPH, with Average Occupancy:29.257435

COMPLETED 120 SECONDS OF SIMULATION..... (CORSIM) ( 3.28 CPU Seconds)
COMPLETED 180 SECONDS OF SIMULATION..... (CORSIM) ( 3.14 CPU Seconds)
COMPLETED 240 SECONDS OF SIMULATION..... (CORSIM) ( 3.27 CPU Seconds)
COMPLETED 300 SECONDS OF SIMULATION..... (CORSIM) ( 3.40 CPU Seconds)
COMPLETED 360 SECONDS OF SIMULATION..... (CORSIM) ( 3.33 CPU Seconds)
COMPLETED 420 SECONDS OF SIMULATION..... (CORSIM) ( 3.21 CPU Seconds)
COMPLETED 480 SECONDS OF SIMULATION..... (CORSIM) ( 3.28 CPU Seconds)
COMPLETED 540 SECONDS OF SIMULATION..... (CORSIM) ( 3.86 CPU Seconds)
COMPLETED 600 SECONDS OF SIMULATION..... (CORSIM) ( 3.47 CPU Seconds)
COMPLETED 660 SECONDS OF SIMULATION..... (CORSIM) ( 3.69 CPU Seconds)
COMPLETED 720 SECONDS OF SIMULATION..... (CORSIM) ( 3.76 CPU Seconds)
COMPLETED 780 SECONDS OF SIMULATION..... (CORSIM) ( 4.09 CPU Seconds)
COMPLETED 840 SECONDS OF SIMULATION..... (CORSIM) ( 4.28 CPU Seconds)
COMPLETED 900 SECONDS OF SIMULATION..... (CORSIM) ( 6.30 CPU Seconds)
TOTAL CPU TIME FOR THIS RUN = 134.97 SECONDS
**** THERE WERE 1 WARNING MESSAGES.

```

Figure 5-6. Screen shot of text outputting to screen when speed changes

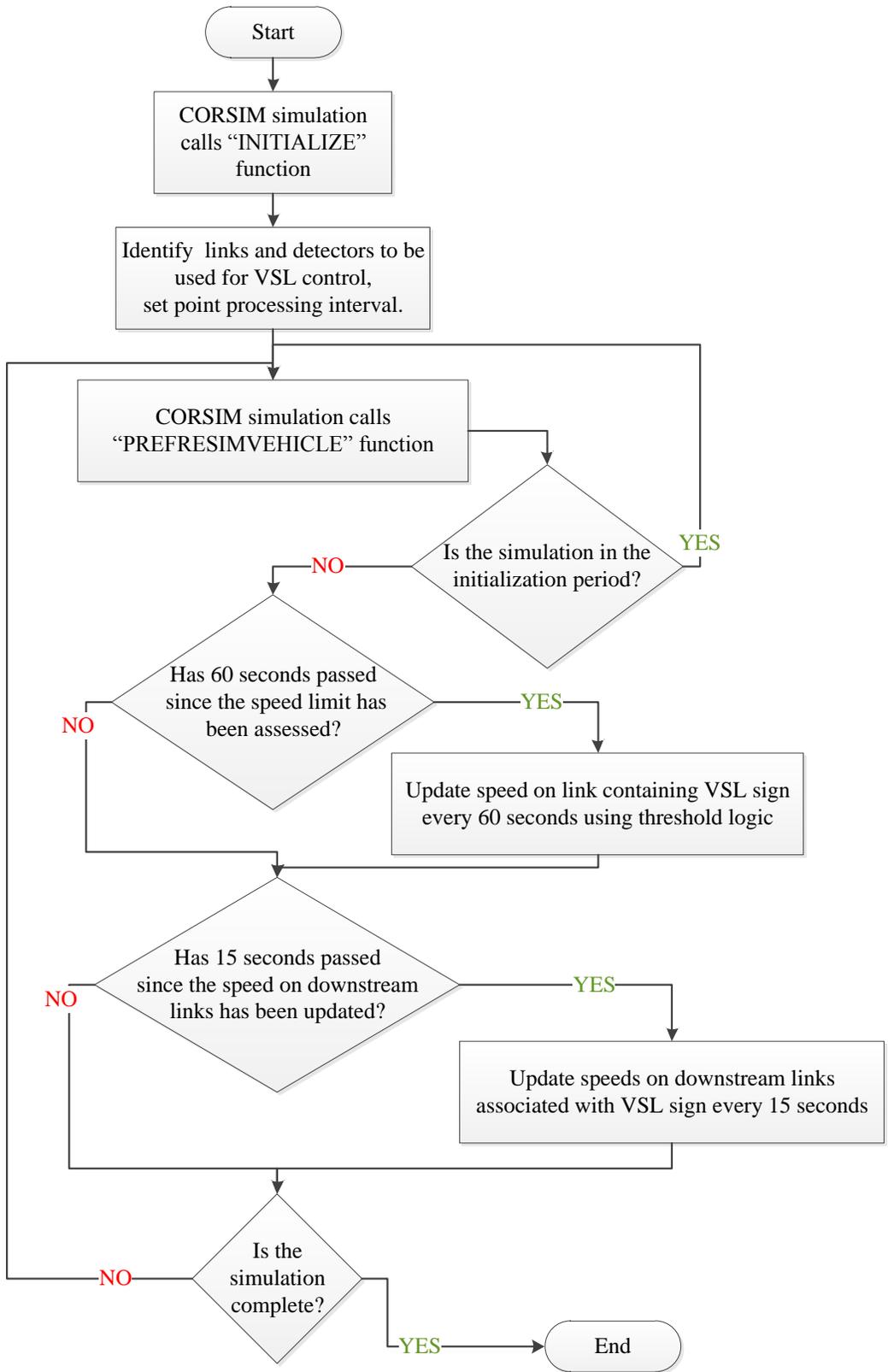


Figure 5-7. Flowchart of general RTE logic

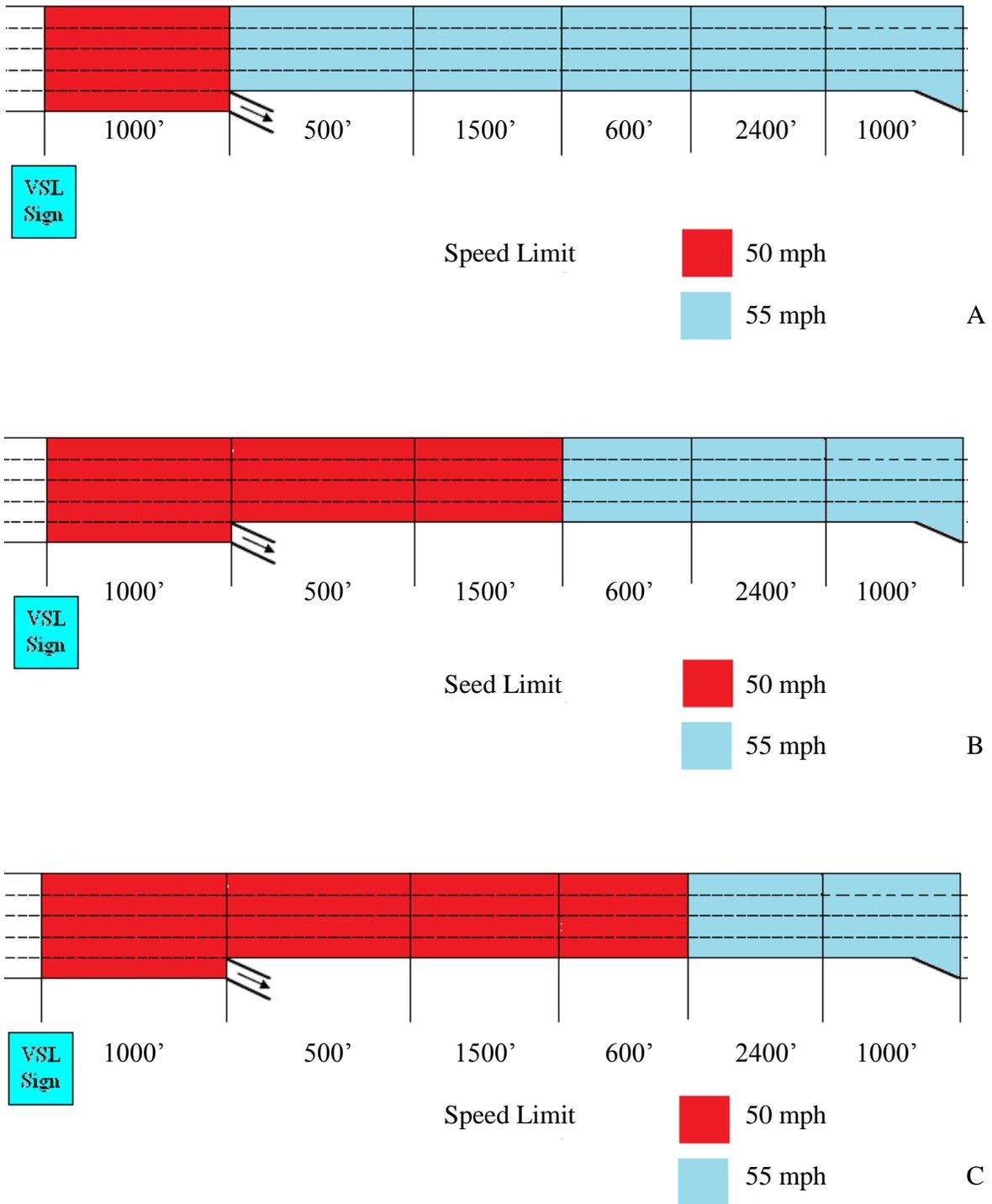
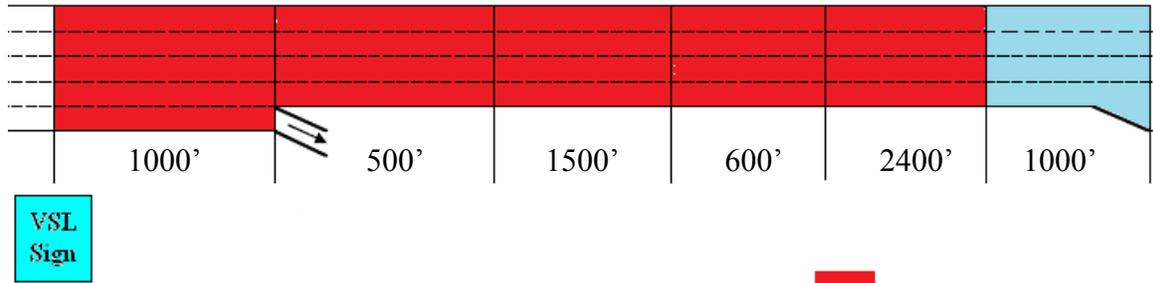
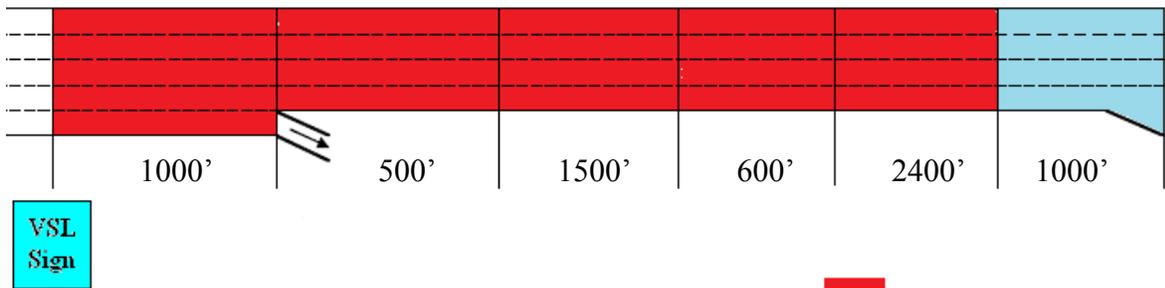


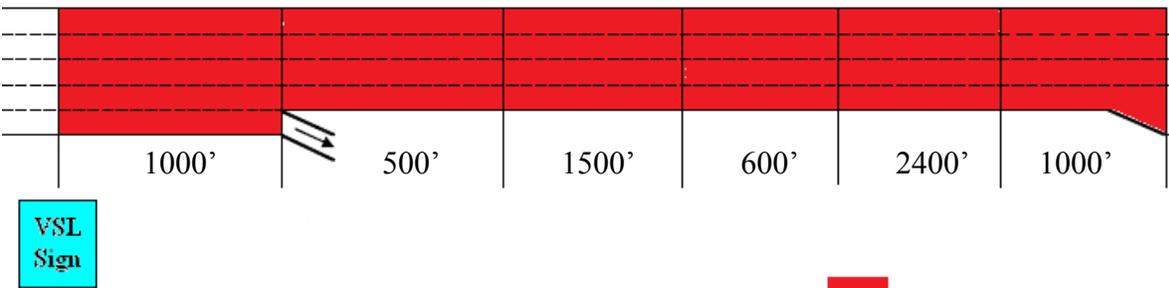
Figure 5-8. Time series of speed limit propagation downstream from VSL sign location. A) 0 seconds. B) 15 seconds. C) 30 seconds. D) 45 seconds. E) 60 seconds F) 75 seconds



Speed Limit ■ 50 mph
■ 55 mph D



Speed Limit ■ 50 mph
■ 55 mph E



Speed Limit ■ 50 mph
■ 55 mph F

Figure 5-8. Continued

CHAPTER 6 IMPLEMENTATION AND ANALYSIS OF ALGORITHMS IN CORSIM

This chapter presents the implementation of the algorithms and results obtained from the simulations. Section 6.1 describes all the scenarios that are implemented and how the simulations are carried out. Section 6.2 examines the results of the no-control scenario, and identifies the bottleneck source. Section 6.3 provides a comparison between the thresholds and sign positioning for each algorithm. Section 6.4 provides a comparative analysis between the best performing scenarios among the three algorithms. The chapter concludes with a summary of the findings.

6.1 Implementation of RTE Scenarios in CORSIM

A different run time extension (RTE) is created for each algorithm and a total of 24 scenarios are developed and tested. The description of each scenario is shown in Table 6-1. The RTEs are loaded into the network by adding a new tool to the tool configuration list. The CORSIM Driver is specified as the tool to run the simulation, and the associated extension is a “trf” file. This process is shown in Figure 6-1. The RTE dynamic link library (DLL) is then loaded and the associated call points are specified. For the RTE created the only call points are the INITIALIZE, and PREFRESIMVEHICLE call points. This means that the RTE will interface with CORSIM at these times. This is shown in Figure 6-2.

Each scenario’s multiple run properties page is set up identically. Initially ten runs were conducted from the no-control scenario to obtain the final number of runs needed. The number of runs required for each scenario was based on the average speed of vehicles in the network in miles per hour. The standard deviation of the initial ten runs was 0.2865. The acceptable error was set to be 0.2 mph, and a confidence interval of 95% was used. The acceptable number of

runs computed was approximately eight. Since originally ten runs were used and this value exceeded the minimum of eight, ten runs are used for all scenarios.

Output processing is performed to provide averages over the ten runs. The simulation generates a comma separated value file that aggregates evaluation parameters by time period. On the network level the parameters include average travel speed, vehicle miles traveled, total travel time, and throughput. On the link level speed profile plots are created, displaying the average speeds over a 3 mile section of roadway upstream from the bottleneck location. Each scenario consists of twelve time periods, fifteen-minutes each. This is a total of 3 hours of simulation time for each scenario. Each simulation requires 3000 seconds of initialization period to properly load the network with vehicles.

6.2 Analysis of the No-Control Scenario

The speed profile of the no-control scenario is displayed in Figure 6-3. From the results it is clear that a bottleneck forms on link 159-161, beginning sometime between time period 3 and 4. During time period 3 the speed at this location has dropped to 35 mph, and at time period 4 it has dropped below 25 mph. The congestion moves upstream and affects links as far as link 147-148, which is over 2 miles upstream of the bottleneck source. Congestion does not dissipate until time period 12, which is approximately two hours after the initial breakdown. This represents a typical evening peak period with recurring congestion on the I-95 network, and an ideal scenario to test the selected variable speed limit (VSL) algorithms.

6.3 Analysis of VSL Algorithm Performance

This section examines each algorithm individually based on network and link statistics. The speed profile and throughput plots are observed for each of the scenarios tested. The effectiveness of the different sign location scenarios is also evaluated.

6.3.1 Analysis of the Occupancy-Based Algorithm

A total of twelve occupancy-based scenarios are simulated. This includes three sets of thresholds, and four different sign configurations. The network-wide performance measures are shown in Table 6-2. Based on these, the scenario that shows the greatest improvement in traffic conditions is using two signs spaced one half mile apart and using the threshold scenario 2 (based on NCHRP Report 3-87). The scenario 2 thresholds are shown in Table 6-3.

This scenario displays the greatest improvement in network average travel speed with a 3.38 % (1.34 mph) increase in average travel speed. The total travel time is decreased by 3.35% (255 hours), which is also the greatest among the occupancy scenarios. The speed profile for this scenario compared to the no-control scenario is shown in Figure 6-4.

The figure shows that before the breakdown the VSL control scenario results in a slight reduction in speeds. This is because the algorithm is reducing speeds below the free-flow speed. During congested conditions the speed difference is negligible between the no-control and control scenarios. However, starting at time period 5 a noticeable improvement in speed is observed starting at the bottleneck source and progressing upstream. This trend continues, moving upstream and increasing average speeds by as much as 16 mph. There is minimal improvement observed at the bottleneck, but significant improvements can be observed in the upstream links. It appears that the speed of the shockwave has been reduced, reducing the effects of congestion on upstream links. It also appears that congestion does not extend as far upstream as in the no control scenario. The length of queue is shortened by approximately 1/3 of a mile.

To evaluate the throughput of the section, scaled cumulative curves were constructed to provide the throughput at key locations around the bottleneck. The scaled cumulative departures are used because they can show much more clearly the differences between the control and no-control case. The scaled cumulative departures are obtained by subtracting the time multiplied

by a base flow rate from the cumulative departures. This relationship is shown in equation 6-1 below:

$$\textit{scaled cum. departures} = \textit{cum. departures} - \textit{base flow rate} * \textit{time period} \quad (6-1)$$

The scaled cumulative departures are obtained at three links along the section. The first area observed is at link 146-147. This is the farthest point upstream where the bottleneck has affected traffic conditions, and a base flow rate of 6,840 vehicles per hour was used as the scaling factor. The next area observed is at link 153-154. This is in the middle of the section affected by the bottleneck, and a base flow rate of 6,280 vehicles per hour was used as the scaling factor. The final location observed is at link 159-161. This is at the entrance to the bottleneck, and a base flow rate of 8,120 vehicles per hour was used as the scaling factor.

Figure 6-5 shows the throughput of this scenario compared to the no-control case. The vertical axis represents the scaled cumulative departures (vehicles), and the horizontal axis represents the time period. The VSL control scenario shows increased throughput over the duration of the congestion, and at every location affected by the bottleneck. As shown, the throughput increases by as much as 92 vehicles for a given 15-minute period.

6.3.2 Analysis of the Volume-Based Algorithm

A total of eight volume-based scenarios were tested. This includes two sets of thresholds, and four different sign configurations. The total network-wide performance measures are shown in Table 6-4. There is no single scenario that can be identified as having the best performance with respect to all measures. There are two scenarios that perform best in specific measures. The first one uses one sign spaced one mile from the bottleneck, and uses the threshold scenario 1 (based on the thresholds implemented on the M25 freeway in England, shown in Table 6-5). This scenario displays the greatest improvement in network average travel speed with a 3.64 % (1.44 mph) increase. This scenario also displays the greatest decrease in total travel time by 3.41%

(260 hours). The speed profile for this scenario compared to the no-control scenario is shown in Figure 6-6.

The second scenario produced comparable results to the previous one. It uses the same threshold scenario (scenario 1) but uses two signs spaced ½ mile apart. The scenario displays the third best results for improved average speed and total travel time with changes of 2.67% (1.06 mph) and -2.51% (191 hours) respectively. The speed profile for this scenario compared to the no control scenario is shown in Figure 6-7.

Some differences can be observed in the speed profiles of these two scenarios. Starting with free flow conditions and during the beginning of congestion the one-sign scenario shows slightly reduced speeds, while the two-sign scenario shows a smoother transition of speeds and smaller deviation from the no-control scenario. At the onset of congestion both scenarios mirror the no control scenario almost identically. As time progresses both scenarios show improved speeds starting at the bottleneck and moving upstream. The first scenario shows consistent average speed improvements by as much as 13 mph. The second scenario shows a similar pattern with speeds increasing by as much as 16 mph. During the recovery phase the one-sign scenario shows greater speed improvement than the two-sign scenario, though both show an improvement over the no-control scenario. Other scenarios from this algorithm showed similar improvements in average speed increase, and travel time reduction.

The throughput for these two scenarios is shown in Figure 6-8 and Figure 6-9. Both scenarios showed improvements in throughput over the duration of congestion. This improved throughput accounts for the increased average travel speeds, and reduced travel times. This finding was observed in all the volume-based scenarios. While both scenarios discussed display improved throughput, the scenario using two signs spaced ½ mile apart shows greater

improvement in terms of throughput by displaying increases of as much as 88 vehicles over a 15 minute period.

6.3.3 Analysis of the Multiple Parameter-Based Algorithm

The parameter thresholds used for the multiple parameter scenarios are determined from the best performing scenarios from the occupancy and volume-based scenarios. Threshold scenario 2 consistently showed the greatest travel time reduction for the occupancy-based algorithm, and scenario 1 consistently showed the greatest travel time reduction in the volume-based scenarios. The resulting threshold conditions for the multiple-parameter algorithm are shown in Figure 6-10.

A similar situation occurred as with the volume-based scenarios, where it is difficult to determine a best case scenario. The scenario with two signs spaced $\frac{1}{2}$ mile from the bottleneck displays the most improvement for average travel speed and total travel time with changes of 2.29% (0.91 mph) and -2.3% (-176 hours) respectively. The speed profile of this scenario compared to the no-control scenario is displayed in Figure 6-11.

The scenario using one sign spaced $\frac{1}{2}$ mile from the bottleneck displays the second best results for average speed and total travel time with changes of 1.08% (0.43 mph) and -1.12% (85.7 hours) respectively. The speed profile for this scenario compared to the no-control scenario is shown in Figure 6-12. From the speed profile of both these scenarios it can be observed that the initial drop in speeds comes much earlier than the no-control case. It appears the drop in speeds caused the traffic breakdown to occur sooner than it normally would have. Through the duration of congestion the control case shows mild improvement over the no-control case. During recovery the speed profile looks nearly identical to the no-control case, and at some points show lower average speeds.

The throughput plots for both these scenarios are shown in Figure 6-13 and Figure 6-14. The scenario using two signs spaced ½ mile apart shows a noticeable increase in throughput by as much as 84 vehicles over a 15 minute period (Figure 6-13). The scenario using one sign spaced ½ mile from the bottleneck showed little improvement if any compared to the no-control case (Figure 6-14).

The scenario with 1 sign spaced 1 mile from the bottleneck produced conditions worse than the no-control scenario for every network parameter. Observing the speed profile in Figure 6-15, the average speed is lower during every time period at the bottleneck location. Also through the duration of the simulation the congestion never dissipates. The throughput for this scenario is shown in Figure 6-16. The throughput for this scenario is less than the no-control scenario over the entire duration of the bottleneck.

6.4 Summary of Analysis

Overall the results show that implementation of VSL increased average speeds and decreased travel times during the simulation. All but one scenario tested show improvement in both of these categories, and these improvements were clearly seen when evaluating the study section on a link by link basis. However, the magnitude of the improvement when viewed as an average over the analysis period is very small (a maximum of 3.6% improvement in average travel). In general, the throughput over the section affected by the bottleneck showed improvement over the no-control scenario. This improved throughput explains the improvement in average travel speed and reduced travel time. In these simulations the volume-based algorithm shows the best overall improvement. The multiple parameter algorithm shows the least improvement overall, and one of its scenarios shows worsening of conditions compared to the no-control scenario. One of the possible explanations for why the volume-based algorithm performs better than the occupancy-based one is that occupancy remains relatively flat over a

wide set of volumes; thus the volume-based algorithm is quicker to trigger a speed limit drop upstream. With respect to the multiple parameter algorithm, it appears that it results in breakdown earlier than in the no-control case. Perhaps this is related to the particular thresholds used, and would not occur for higher thresholds. Note that this analysis has not sought to thoroughly evaluate and compare these algorithms. To completely assess each algorithm and document their relative merits and preferred applications, a full optimization of thresholds and sign positioning would have to be performed.

A comparison can be made between the use of one sign and the use of two signs. Using only one sign creates a sharp transition between speeds and results in lower speeds before the onset of congestion. Using two signs creates a smoother transition between speeds and as a result displays less reduction in speeds before the onset of congestion. However, the best sign spacing is not the same for every algorithm. Each algorithm performs best with different sign spacing.

The key improvement that VSL seems to have is in the speeds upstream of the bottleneck, and in the queue length during congested conditions. It appears that evaluating conditions directly at the bottleneck source will not show the full effect of the VSL system. While conditions are slightly improved at this location for most algorithms, the more significant improvements occur upstream of the bottleneck source.

Also, comparing the average speeds for the entire system shows minimal changes and masks the significant effects of the VSL upstream of the bottleneck. Evaluating effects and considering the entire time and space of the peak period shows benefits up to three miles upstream of the bottleneck source. Average speeds are significantly improved at those locations compared to the no-control case, and queue lengths are significantly reduced. Queue lengths for the occupancy-based algorithm typically show reductions in the range of 0.25 to 0.35 miles. The

volume-based algorithm shows improvements in the range of 0.25 to 0.45 miles. The multiple parameter algorithm showed minimal reduction of queue, but at some points displayed decreases as much as 0.2 miles. Overall network changes appear to be small when conditions are averaged through the entire system, but when considered that this VSL implementation is only affecting two to three miles of a 13-mile stretch, these small changes become more significant.

Table 6-1. Description of scenarios tested

Algorithm	Threshold scenario	Sign location
No VSL control	-	-
Occupancy based	1	One sign - half mile spacing
Occupancy based	2	One sign - half mile spacing
Occupancy based	3	One sign - half mile spacing
Occupancy based	1	One sign - one mile spacing
Occupancy based	2	One sign - one mile spacing
Occupancy based	3	One sign - one mile spacing
Occupancy based	1	Two signs - half mile spacing
Occupancy based	2	Two signs - half mile spacing
Occupancy based	3	Two signs - half mile spacing
Occupancy based	1	Two signs - one-mile spacing
Occupancy based	2	Two signs - one-mile spacing
Occupancy based	3	Two signs - one-mile spacing
Volume based	1	One sign - half mile spacing
Volume based	2	One sign - half mile spacing
Volume based	1	One sign - one mile spacing
Volume based	2	One sign - one mile spacing
Volume based	1	Two signs - half mile spacing
Volume based	2	Two signs - half mile spacing
Volume based	1	Two signs - one-mile spacing
Volume based	2	Two signs - one-mile spacing
Multiple parameter	-	One sign - half mile spacing
Multiple parameter	-	One sign - one mile spacing
Multiple parameter	-	Two signs - half mile spacing
Multiple parameter	-	Two signs - one-mile spacing

Table 6-2. Network performance measures for the occupancy-based scenario

Algorithm	# of signs	Spacing (miles)	Threshold scenario	Average speed	% Change from no-control	Vehicle miles traveled (veh)	% Change from no-control	Total travel time (hours)	% Change from no-control	Total throughput (veh)	% Change from no-control
No control	-	-	-	39.57	0.00	299905.2	0.00	7624.55	0.00	24530.0	0.00
Occupancy	1	1/2	1	40.13	1.41	299987.3	0.03	7514.82	-1.44	24521.6	-0.03
Occupancy	1	1/2	2	40.39	2.05	299803.8	-0.03	7475.28	-1.96	24533.8	0.02
Occupancy	1	1/2	3	40.60	2.60	299640.6	-0.09	7425.28	-2.61	24509.6	-0.08
Occupancy	1	1	1	40.15	1.46	299785.2	-0.04	7512.60	-1.47	24519.6	-0.04
Occupancy	1	1	2	40.82	3.14	299440.8	-0.15	7380.71	-3.20	24475.0	-0.22
Occupancy	1	1	3	40.11	1.35	299888.7	-0.01	7523.84	-1.32	24533.8	0.02
Occupancy	2	1/2	1	40.37	2.02	299782.7	-0.04	7471.69	-2.00	24508.4	-0.08
Occupancy	2	1/2	2	40.91	3.38	300043.3	0.05	7369.21	-3.35	24544.4	0.06
Occupancy	2	1/2	3	40.33	1.90	299602.4	-0.10	7471.87	-2.00	24528.1	-0.01
Occupancy	2	1	1	40.82	3.14	299440.8	-0.15	7380.71	-3.20	19578.2	-20.19
Occupancy	2	1	2	39.89	0.79	299789.3	-0.04	7570.62	-0.71	24501.6	-0.12
Occupancy	2	1	3	40.03	1.15	299710.1	-0.07	7535.28	-1.17	24517.3	-0.05

Table 6-3. Occupancy thresholds for displayed speed limits (scenario 2)

Traffic category	Occupancy for decreasing speed limit (%)	Occupancy for increasing speed limit (%)	Speed limit (mph)
Free flow	< 10	< 8	55
Light congestion	10 - 30	8 - 27	50
Heavy congestion	> 30	>2	45

Table 6-4. Network performance measures for the volume-based algorithm

Algorithm	# of signs	Spacing (miles)	Threshold scenario	Average speed	% Change from no-control	Vehicle miles traveled (veh)	% Change from no-control	Total travel time (hours)	% Change from no-control	Total throughput (veh)	% Change from no-control
No control	-	-	-	39.57	0.00	299905.2	0.00	7624.55	0.00	24530.0	0.00
Volume	1	1/2	1	40.52	2.40	299650.9	-0.08	7445.97	-2.34	24512.9	-0.07
Volume	1	1/2	2	40.36	1.97	299636.7	-0.09	7467.51	-2.06	24497.2	-0.13
Volume	1	1	1	41.01	3.64	300071.9	0.06	7364.17	-3.41	24544.6	0.06
Volume	1	1	2	40.18	1.52	299547.2	-0.12	7498.70	-1.65	24488.6	-0.17
Volume	2	1/2	1	40.63	2.67	300278.7	0.12	7433.17	-2.51	24619.4	0.36
Volume	2	1/2	2	40.39	2.06	299521.1	-0.13	7462.65	-2.12	24492.1	-0.15
Volume	2	1	1	40.86	3.25	299989.8	0.03	7388.90	-3.09	24514.6	-0.06
Volume	2	1	2	39.64	0.18	299799.6	-0.04	7619.88	-0.06	24535.9	0.02

Table 6-5. Network performance measures for the multiple parameter-based algorithm

Algorithm	# of signs	Spacing (miles)	Threshold Scenario	Average Speed	% Change from No-Control	Vehicle Miles Traveled (veh)	% Change from No-Control	Total Travel Time (hours)	% Change from No-Control	Total Throughput (veh)	% Change from No-Control
No control	-	-	-	39.57	0.00	299905.2	0.00	7624.55	0.00	24530.0	0.00
Multiple	1	1/2	1	40.00	1.08	299937.9	0.01	7538.87	-1.12	24520.1	-0.04
Multiple	1	1	1	38.73	-2.14	299544.0	-0.12	7791.58	2.19	24459.5	-0.29
Multiple	2	1/2	1	40.48	2.29	299763.0	-0.05	7448.92	-2.30	24486.0	-0.18
Multiple	2	1	1	39.94	0.92	299433.1	-0.16	7543.46	-1.06	24436.6	-0.38

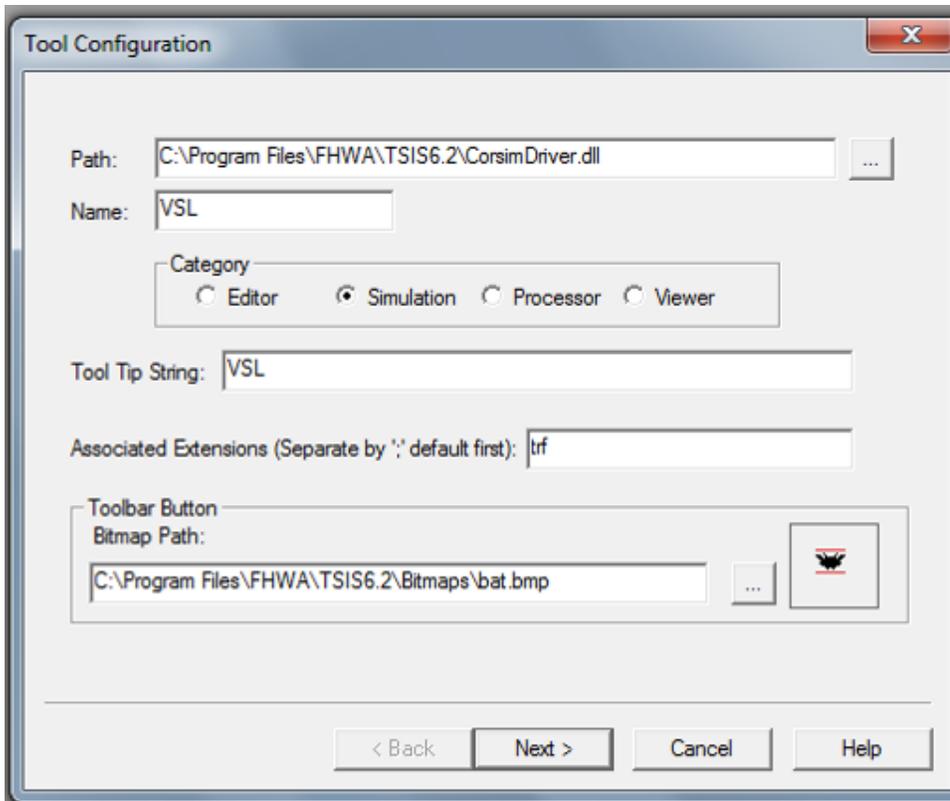


Figure 6-1. Setting up the RTE tool configuration

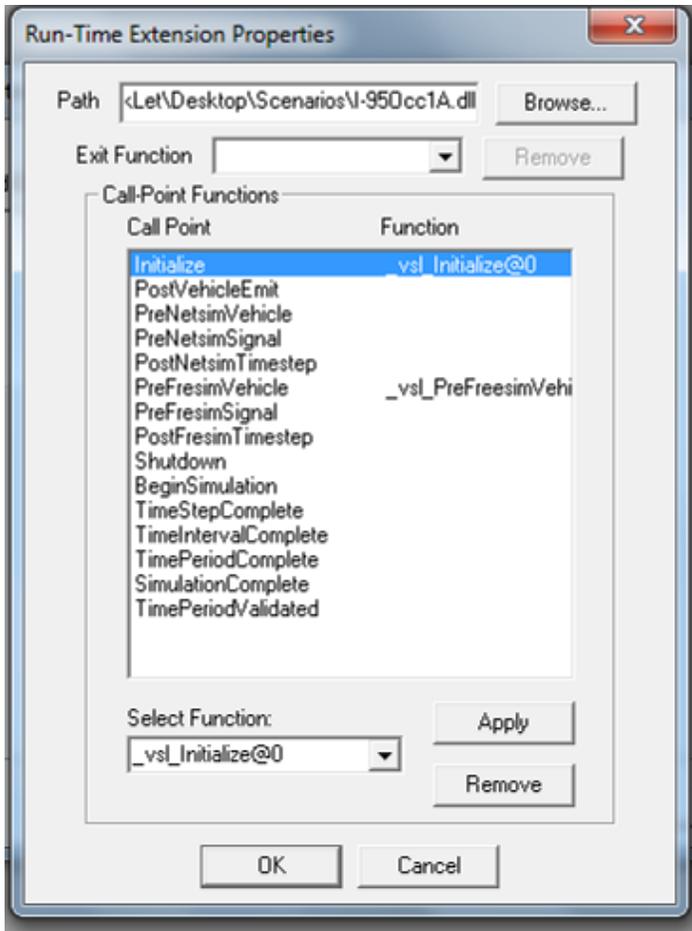
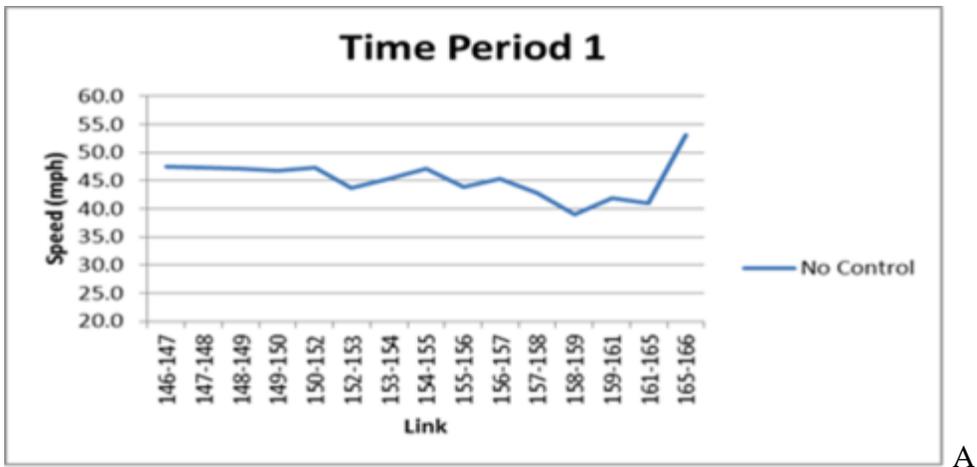
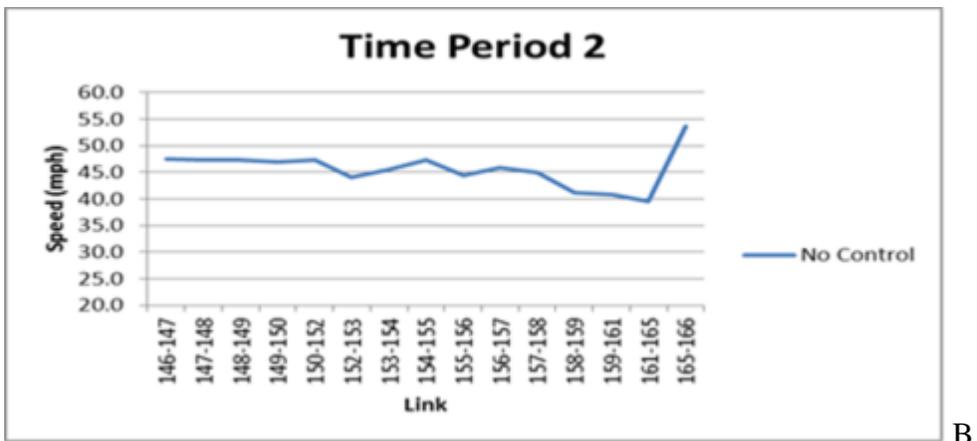


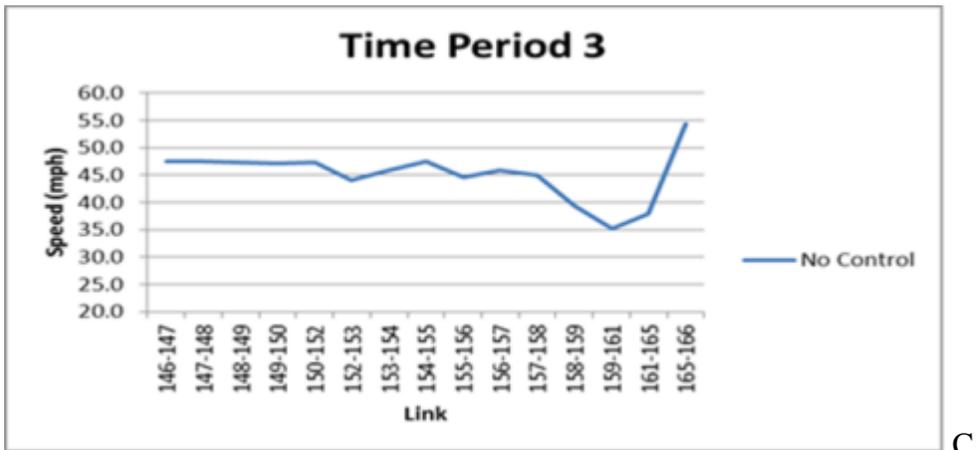
Figure 6-2. Specifying the call points of the RTE



A

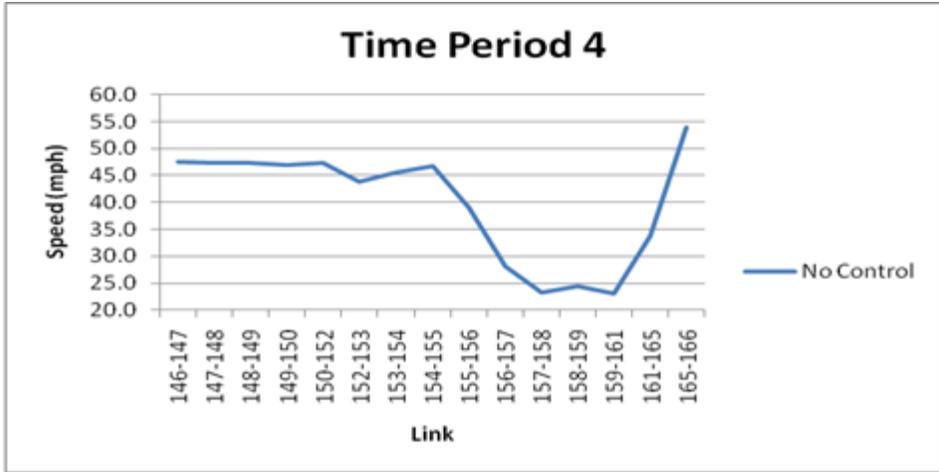


B

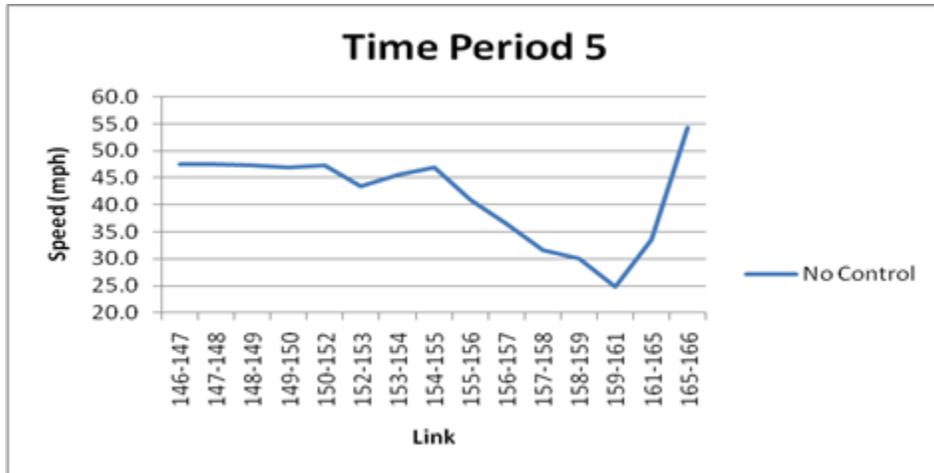


C

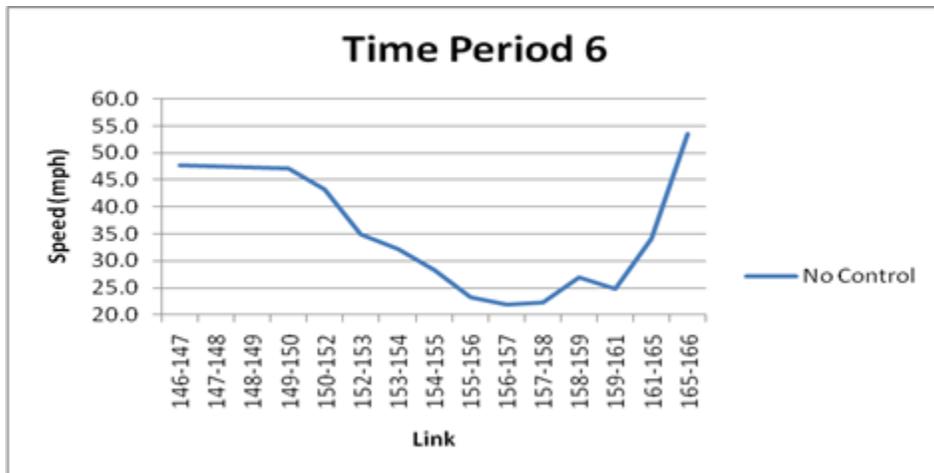
Figure 6-3. Speed profile for no-control scenario over time the 12 periods A) Time period 1. B) Time Period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time Period 6. G) Time Period 7. H) Time Period 8. I) Time Period 9. J) Time Period 10. K) Time Period 11. L) Time Period 12.



D

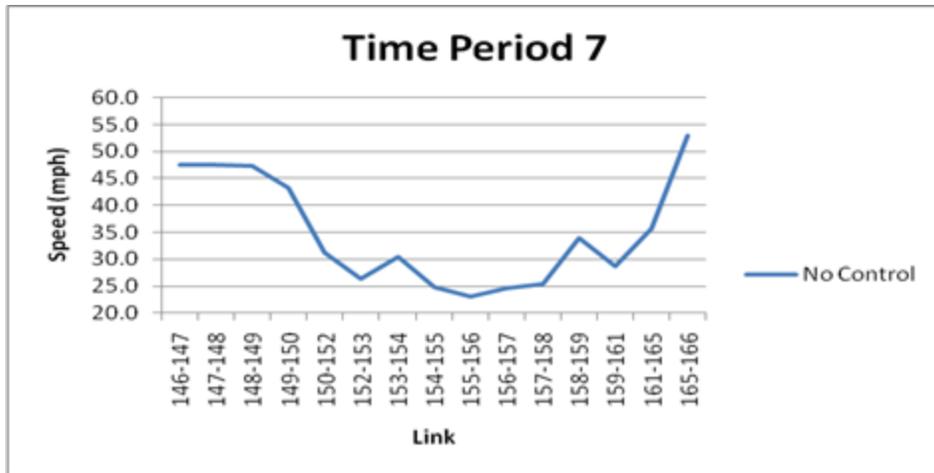


E

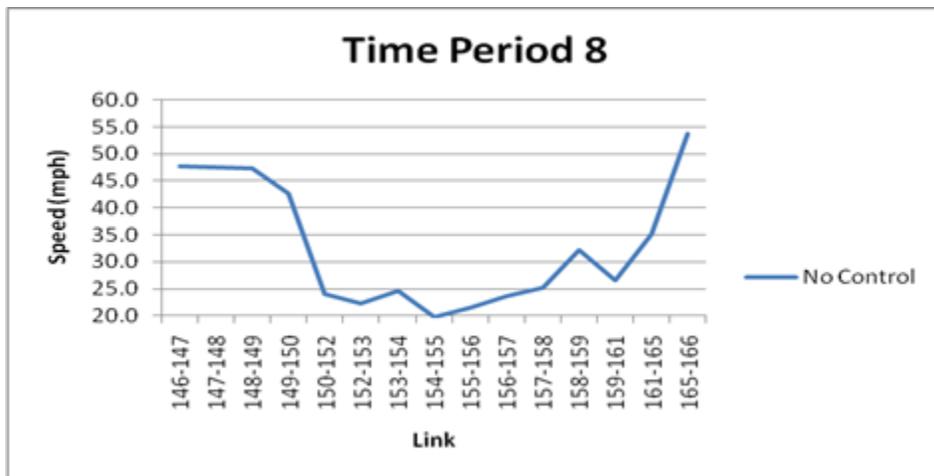


F

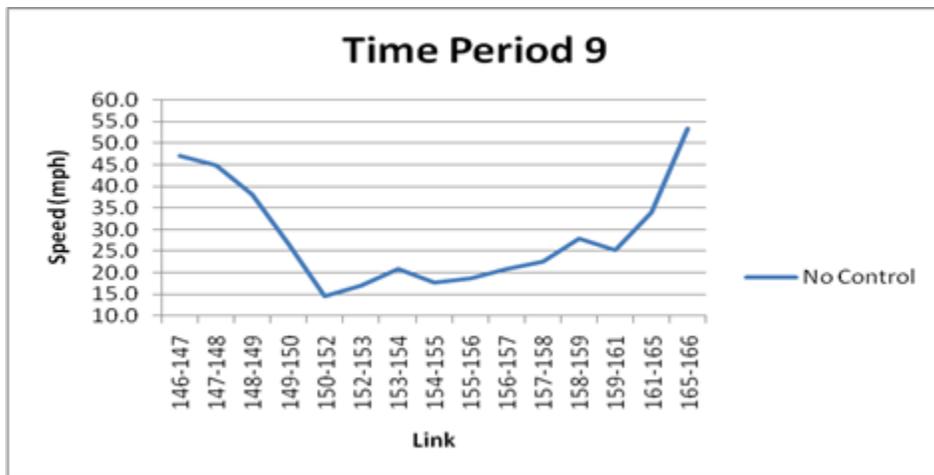
Figure 6-3. Continued



G

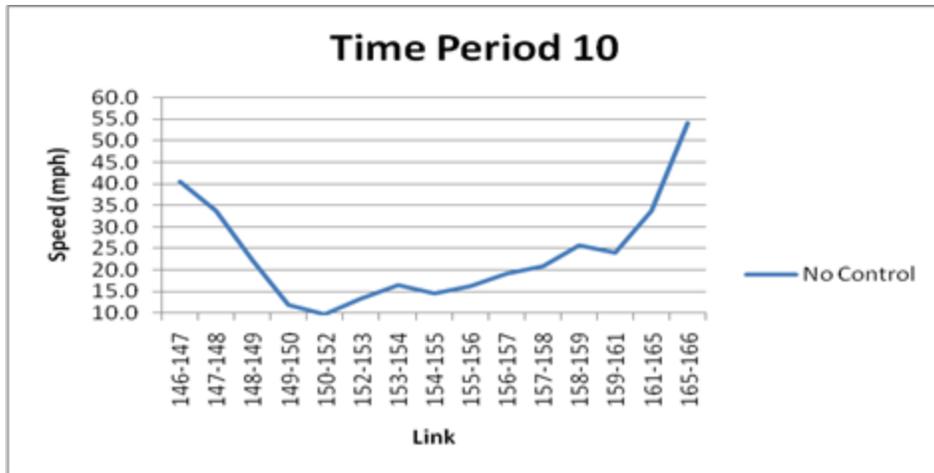


H

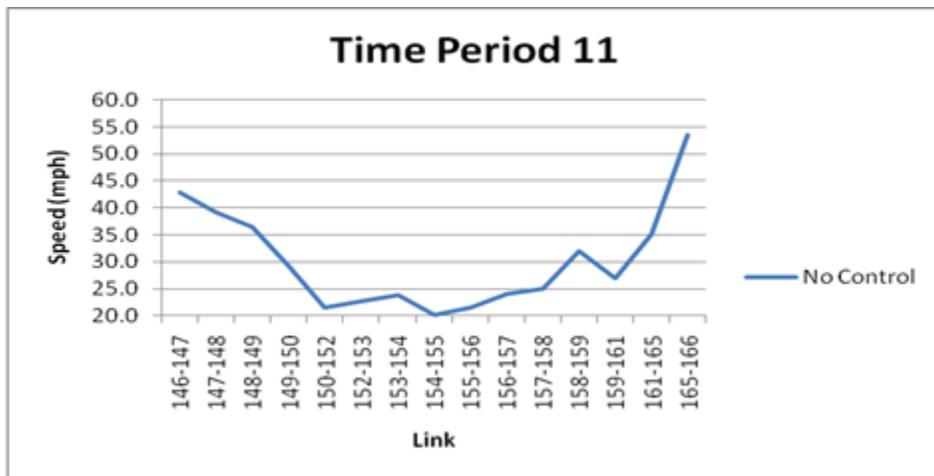


I

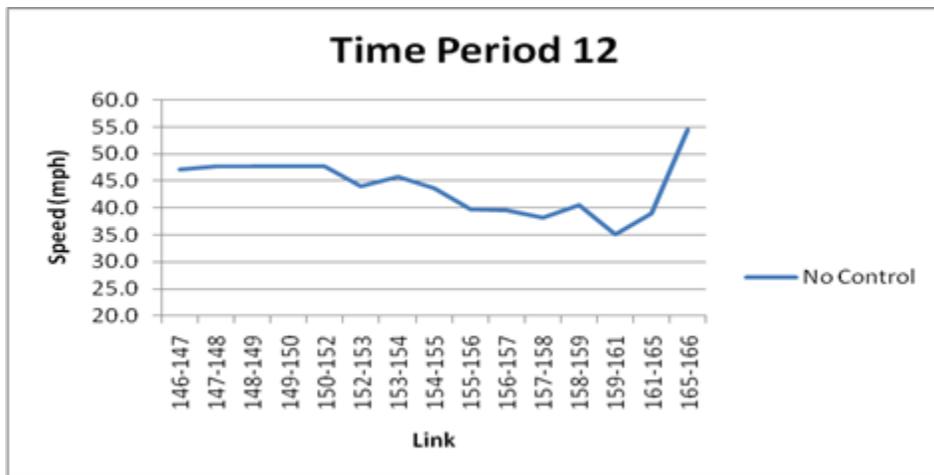
Figure 6-3. Continued



J

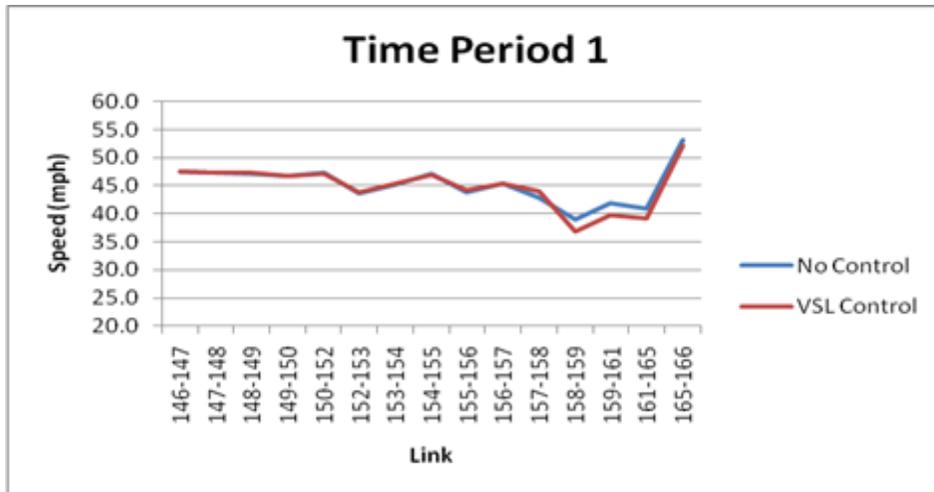


K

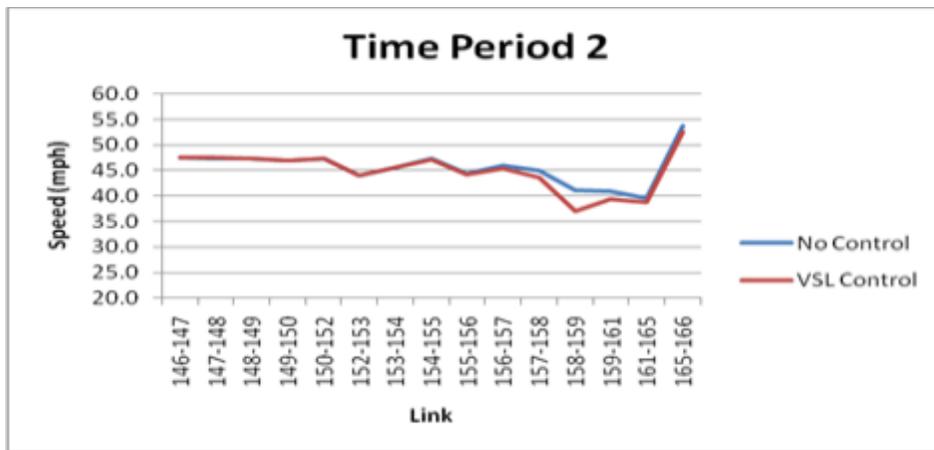


L

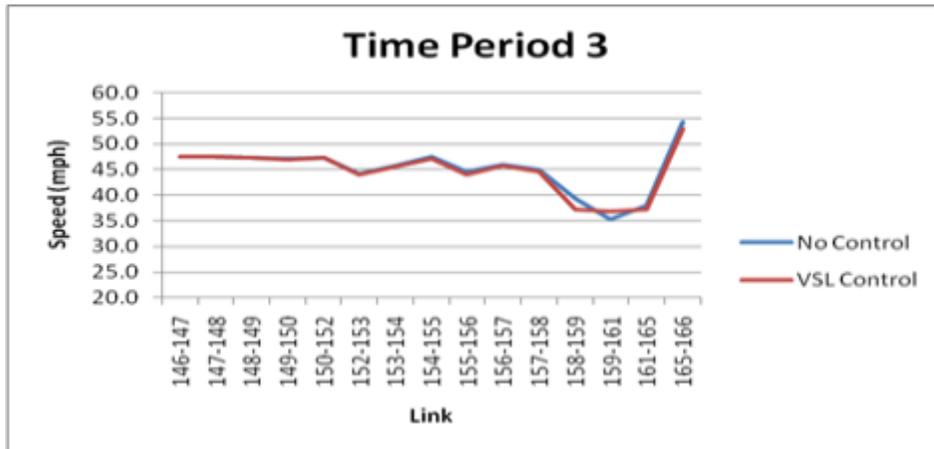
Figure 6-3. Continued



A

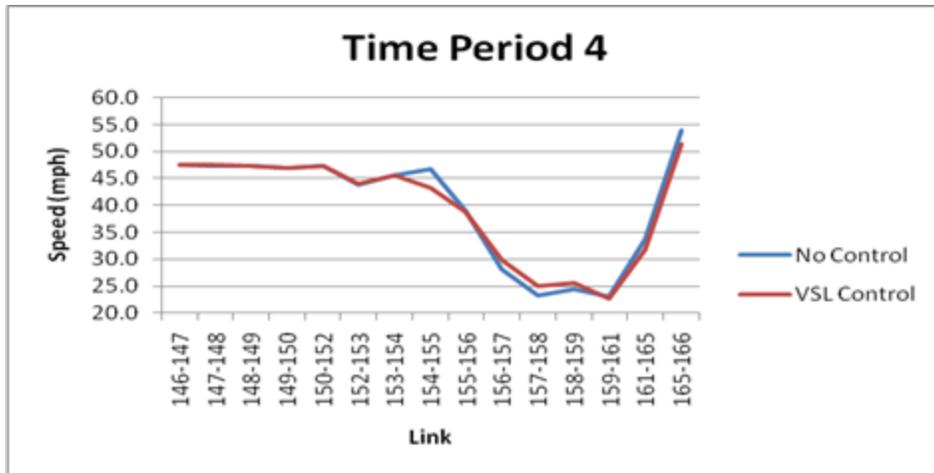


B

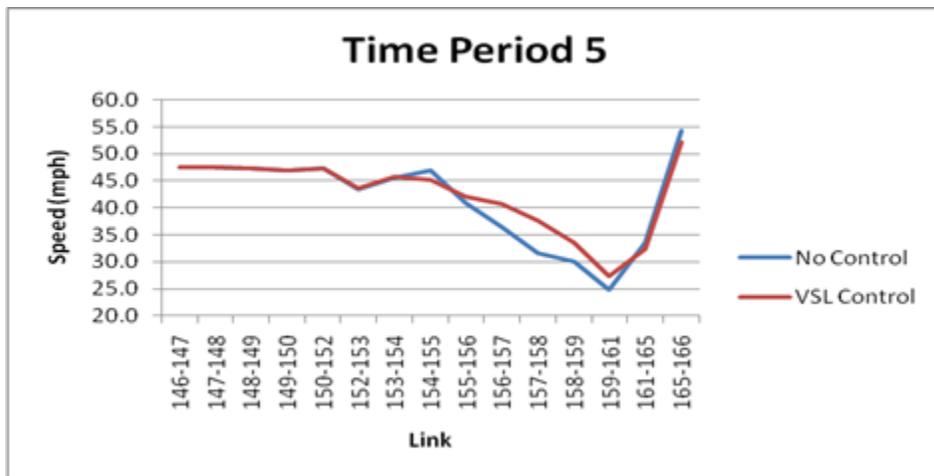


C

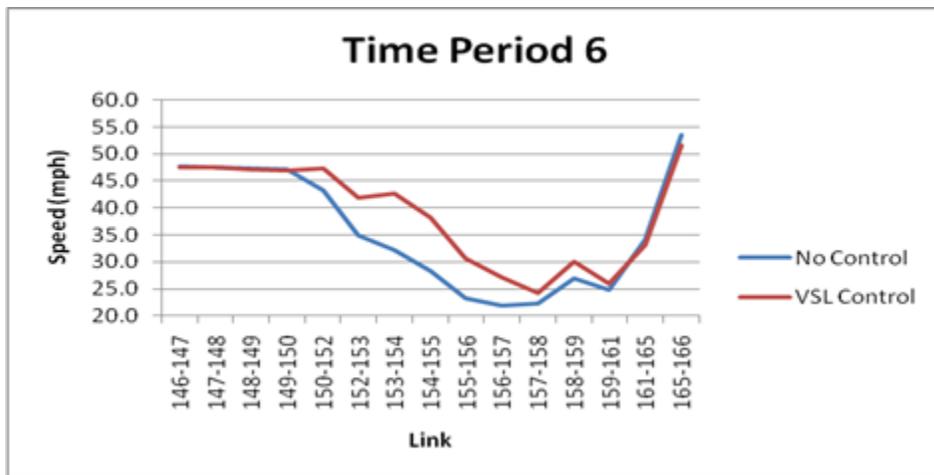
Figure 6-4. Speed profile for the occupancy-based algorithm using two signs spaced ½ mile apart (threshold scenario 2) compared to the no-control scenario A) Time period 1. B) Time Period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time Period 6. G) Time Period 7. H) Time Period 8. I) Time Period 9. J) Time Period 10. K) Time Period 11. L) Time Period 12.



D

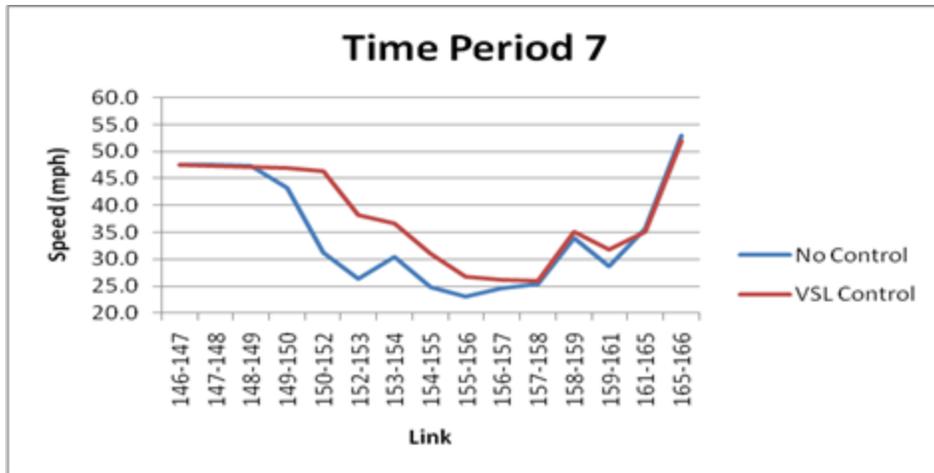


E

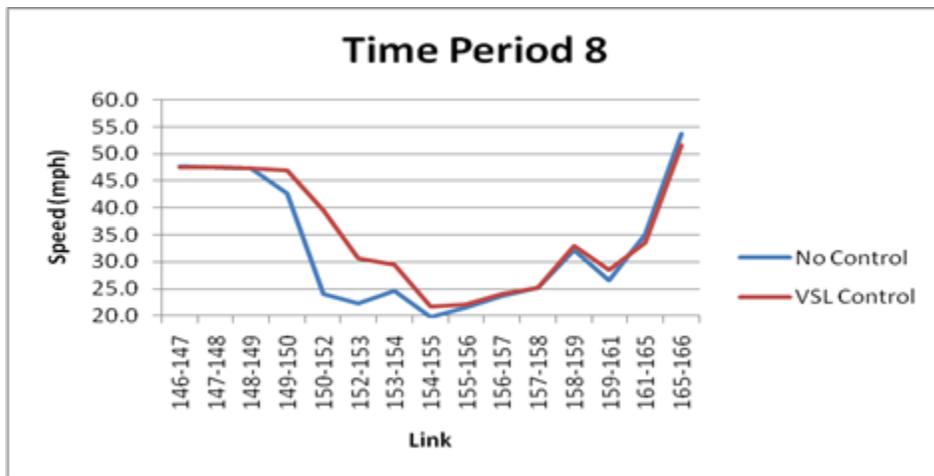


F

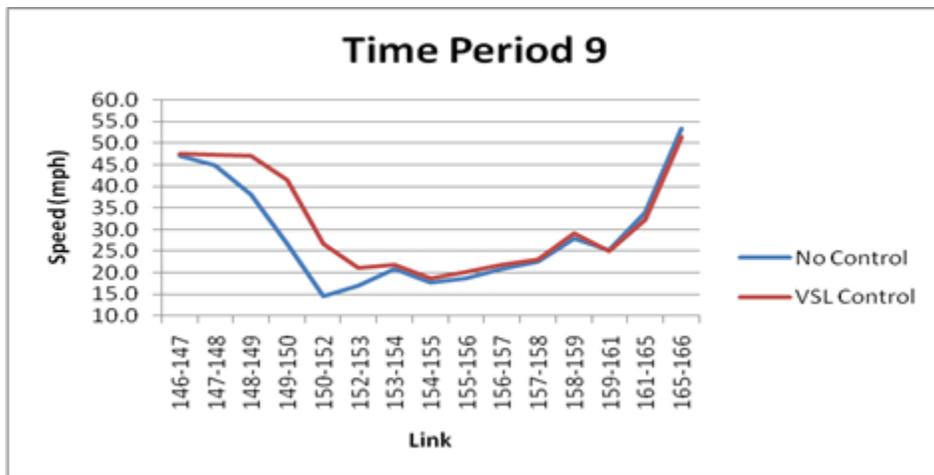
Figure 6-4. Continued



G

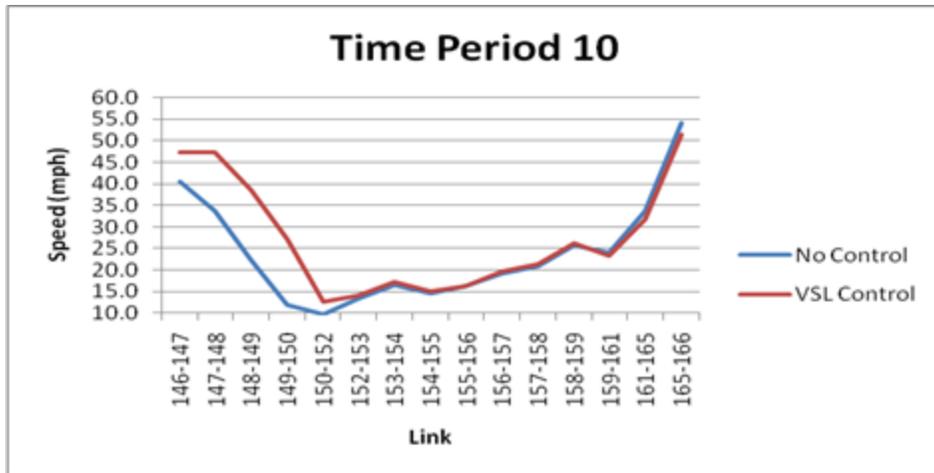


H

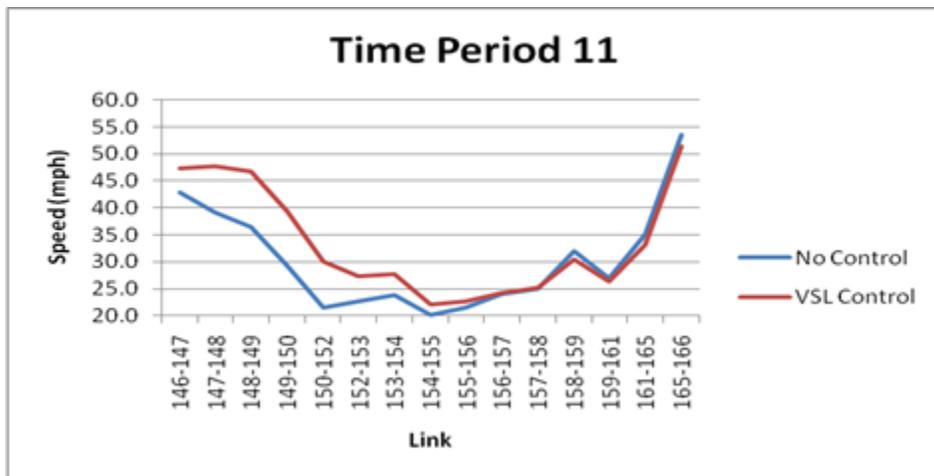


I

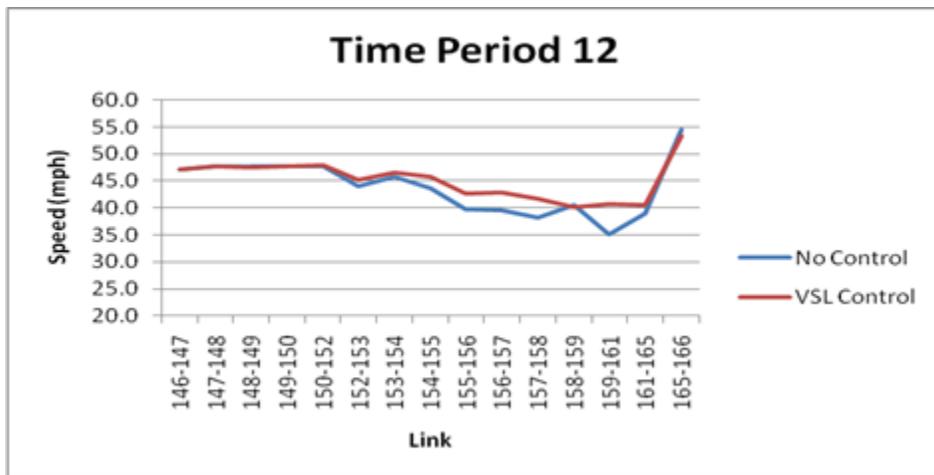
Figure 6-4. Continued



J



K



L

Figure 6-4. Continued

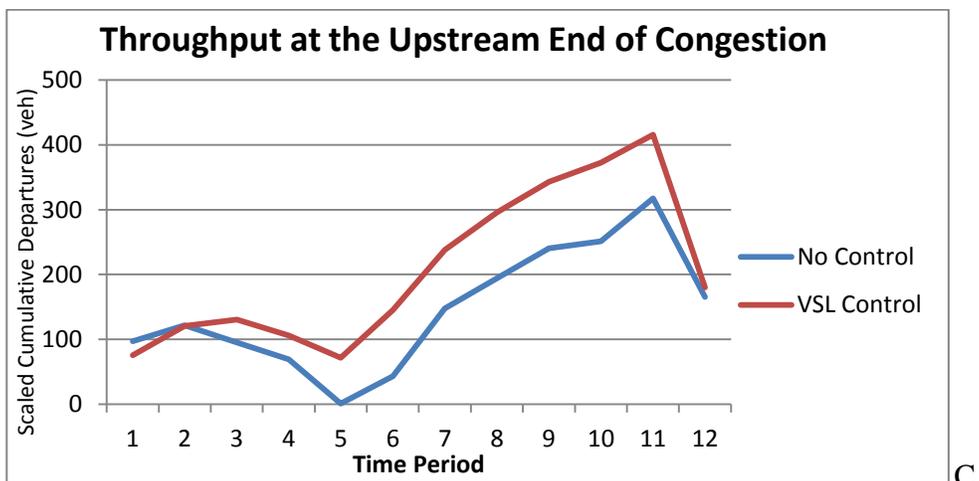
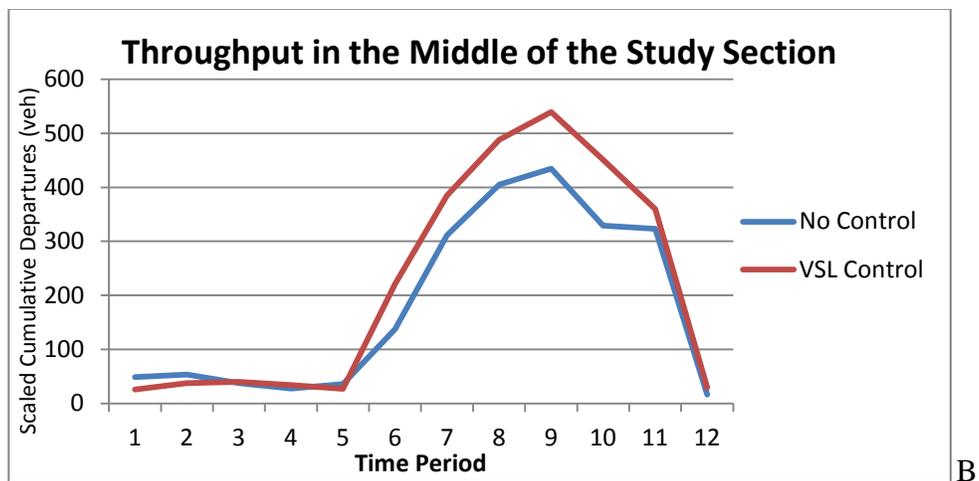
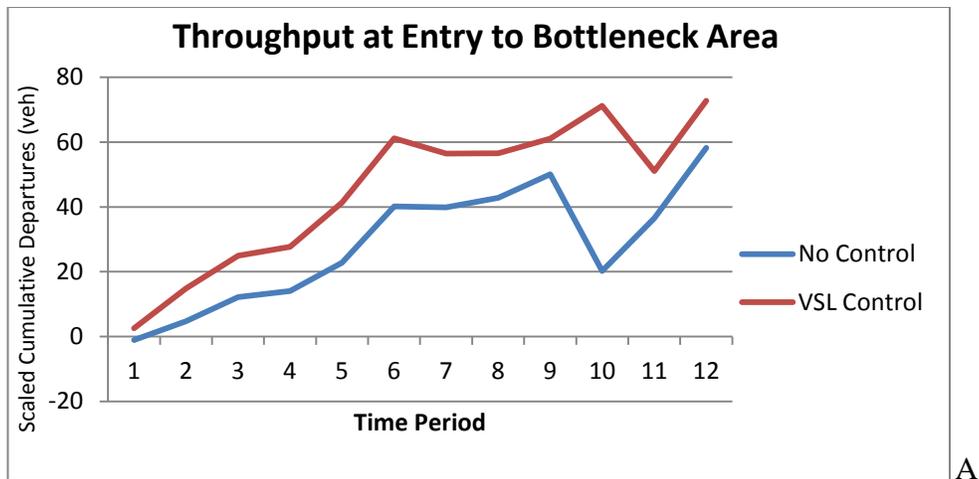


Figure 6-5. Throughput for the occupancy-based algorithm using two signs spaced ½ mile apart (threshold scenario 2) compared to the no-control scenario A) At entry to bottleneck area (link 146-147). B) At the middle of the study section (link 153-154). C) At the upstream end of congestion (link 159-161).

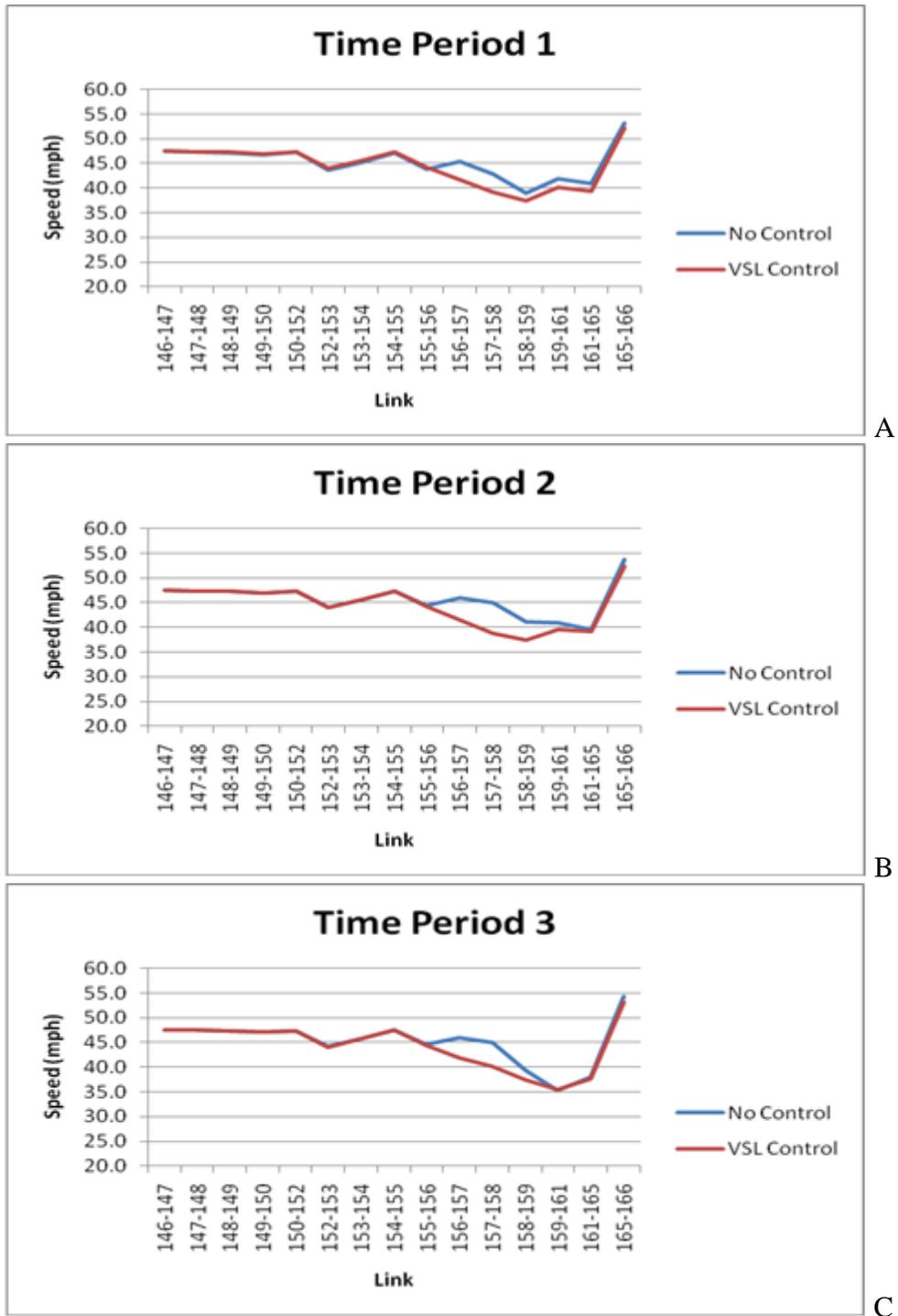
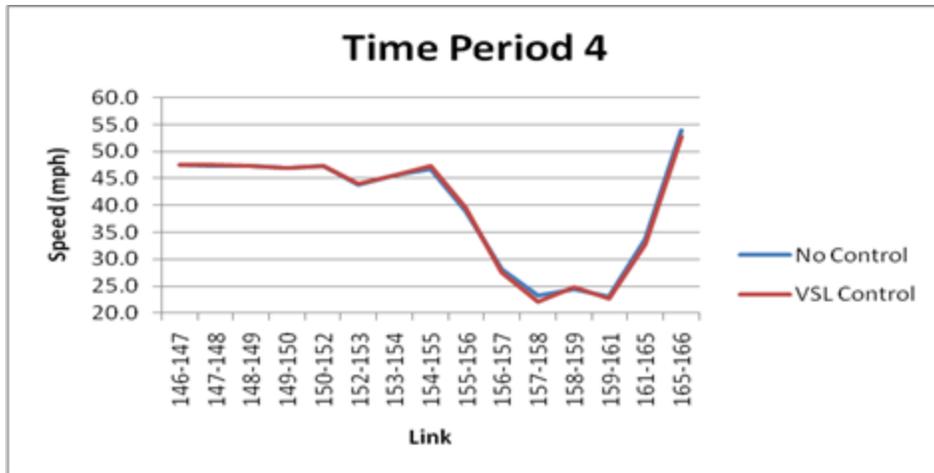
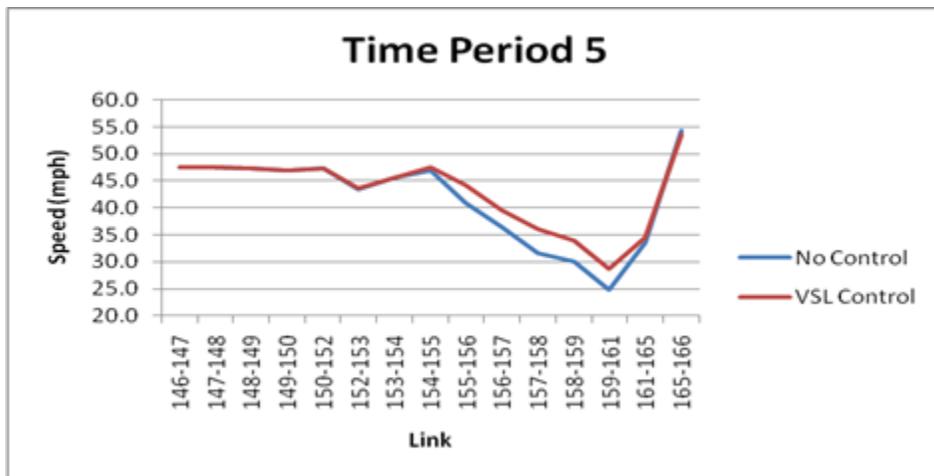


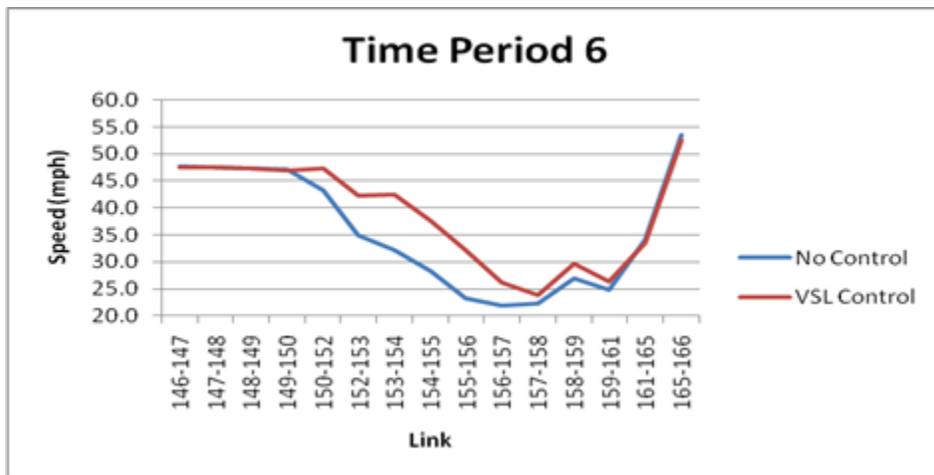
Figure 6-6. Speed profile for the volume-based algorithm using one sign spaced 1 mile from the bottleneck source (threshold scenario 1) compared to the no-control scenario A) Time period 1. B) Time Period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time Period 6. G) Time Period 7. H) Time Period 8. I) Time Period 9. J) Time Period 10. K) Time Period 11. L) Time Period 12.



D

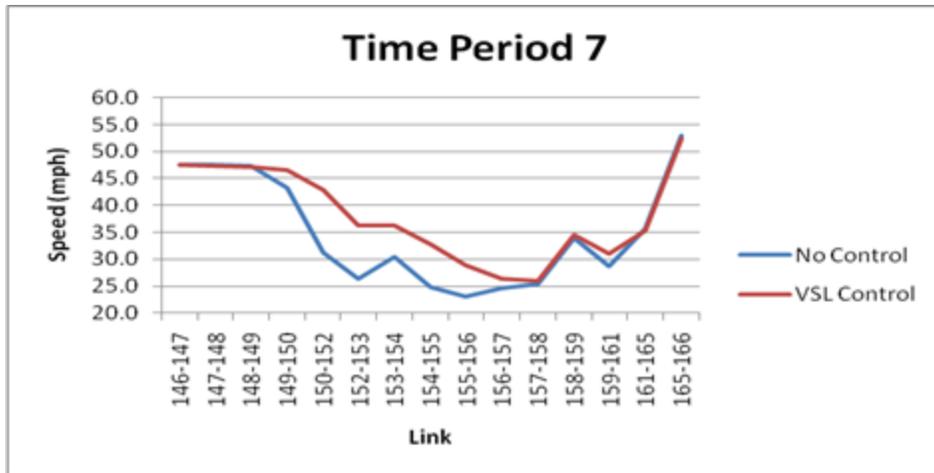


E

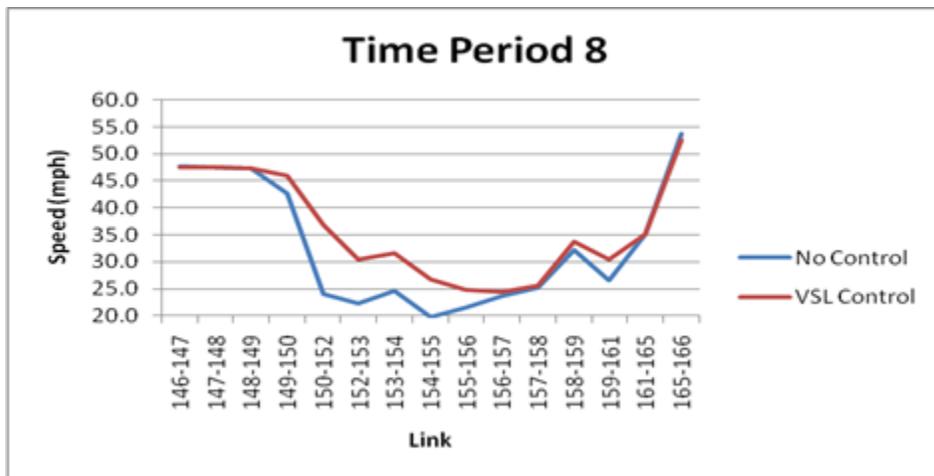


F

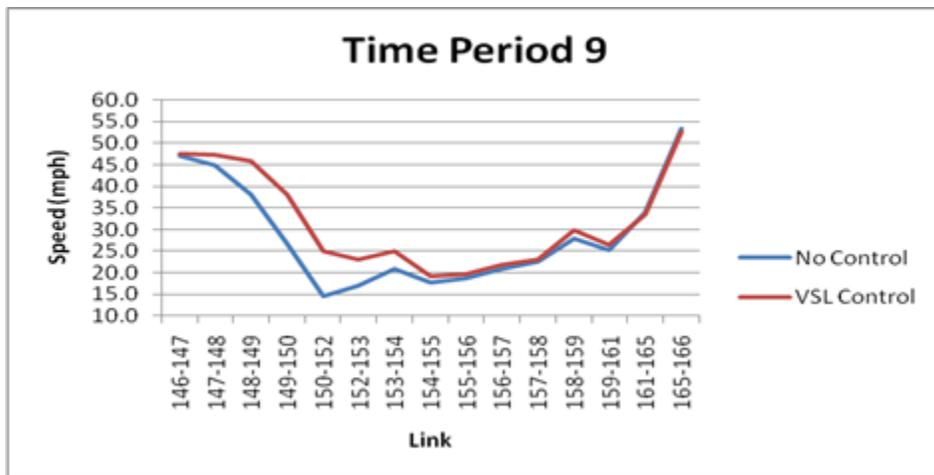
Figure 6-6. Continued



G

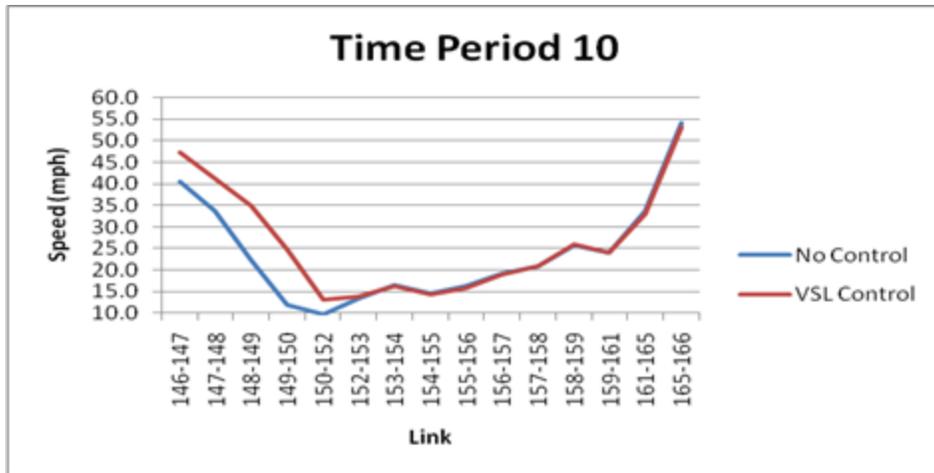


H

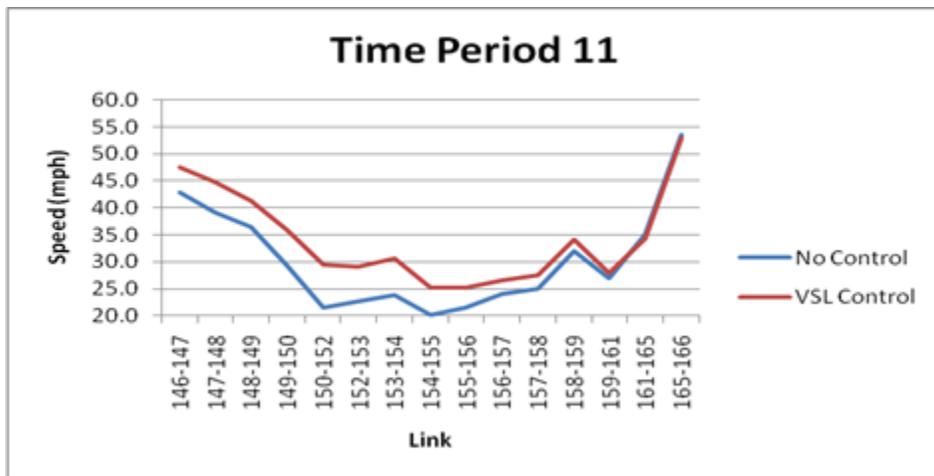


I

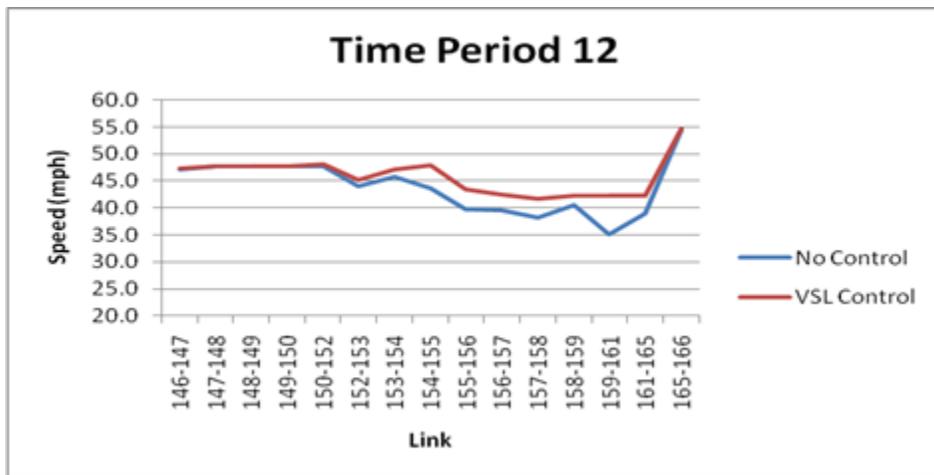
Figure 6-6. Continued



J



K



L

Figure 6-6. Continued

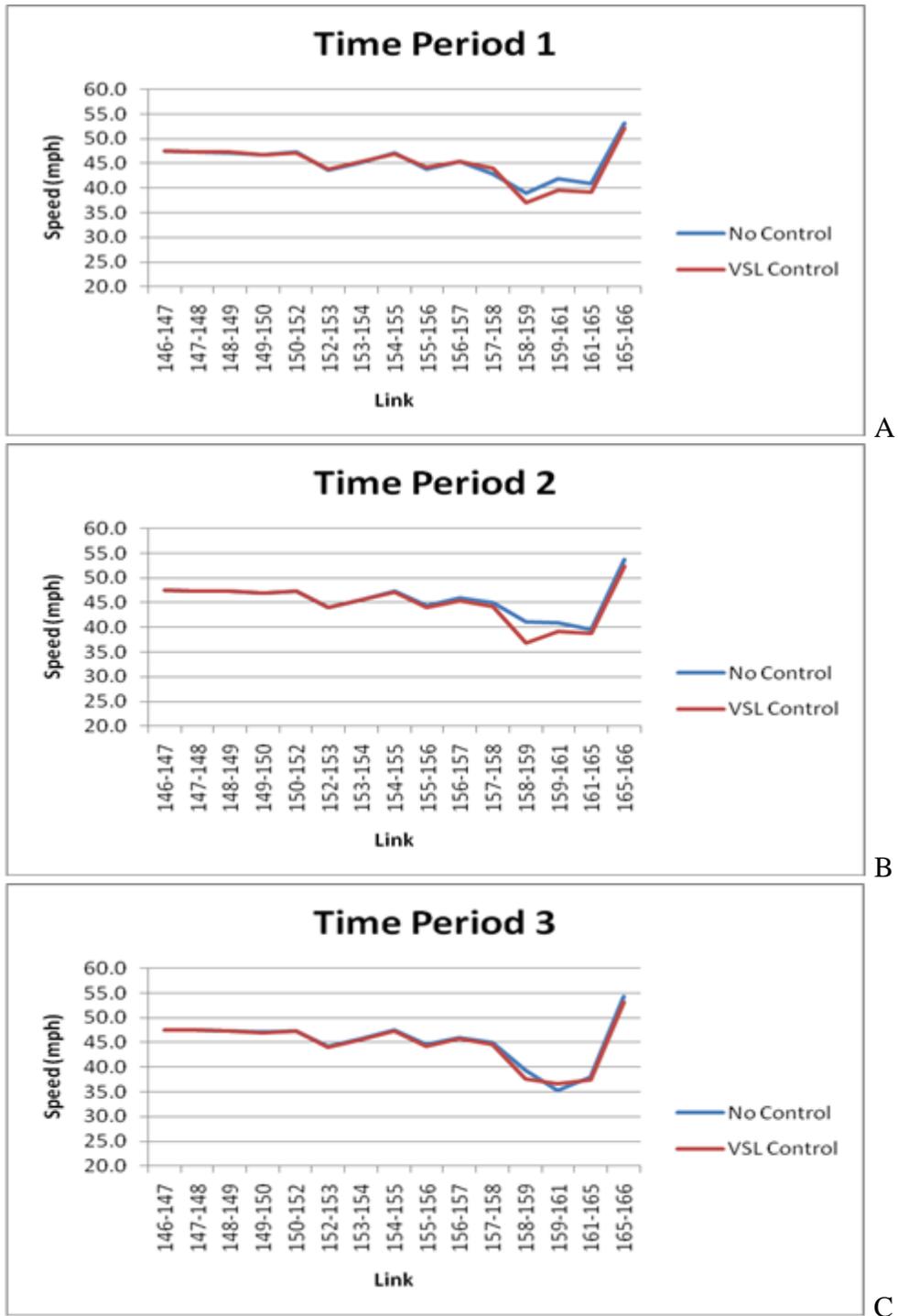
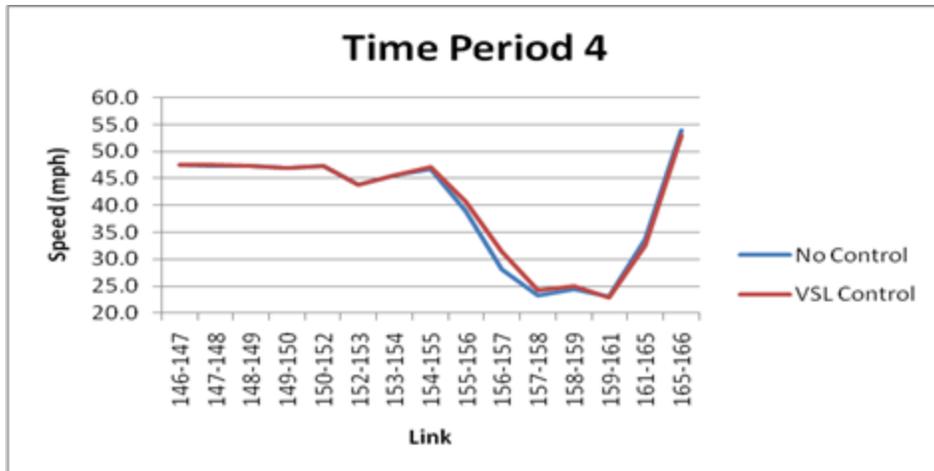
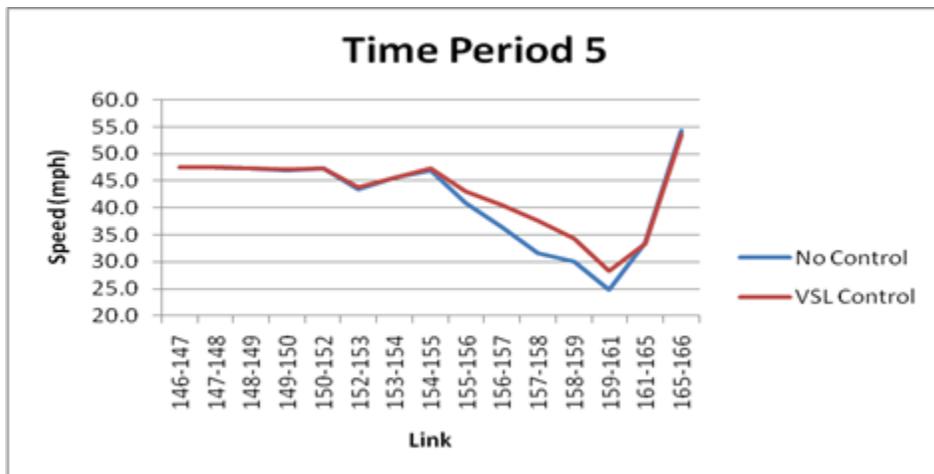


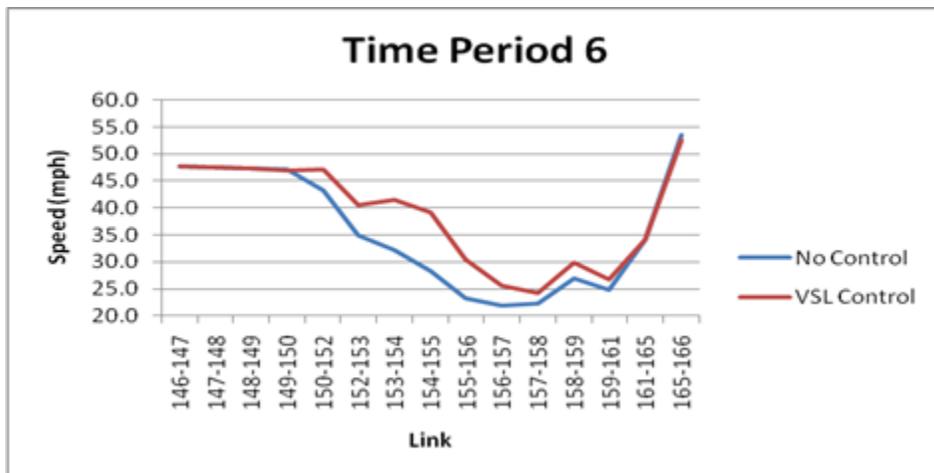
Figure 6-7. Speed profile for the volume-based algorithm using two signs spaced 1/2 mile apart (threshold scenario 1) compared to the no-control scenario A) Time period 1. B) Time Period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time Period 6. G) Time Period 7. H) Time Period 8. I) Time Period 9. J) Time Period 10. K) Time Period 11. L) Time Period 12.



D

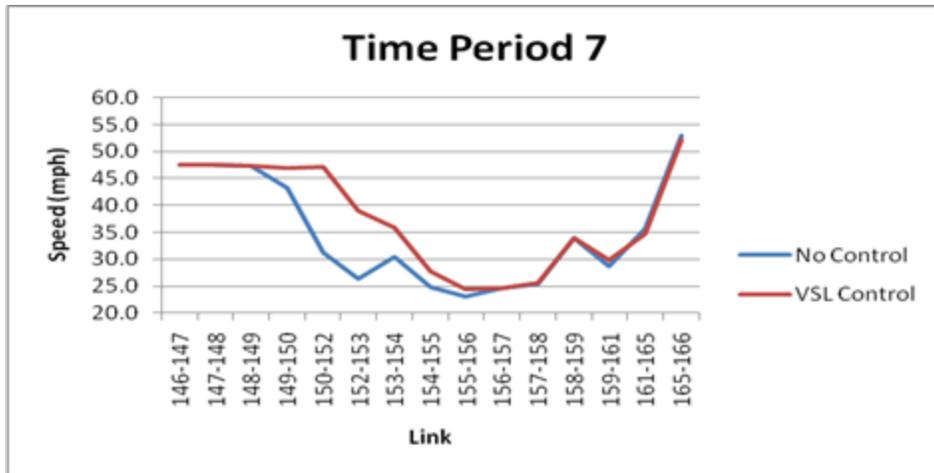


E

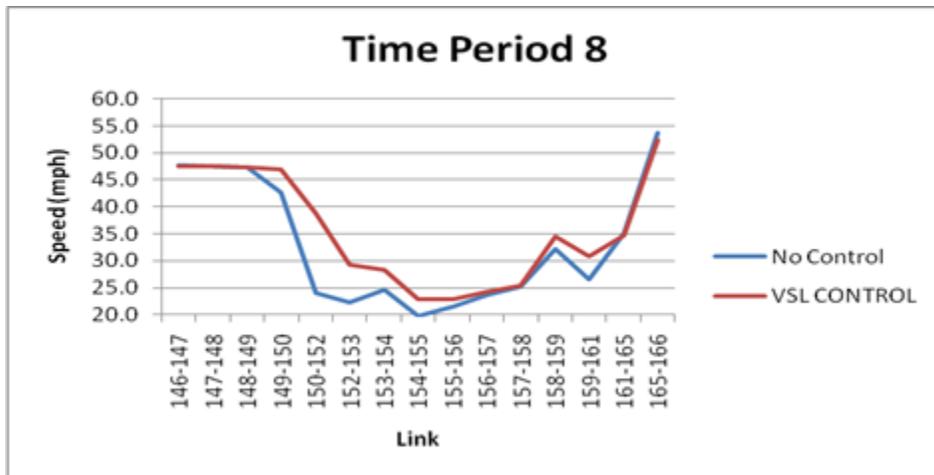


F

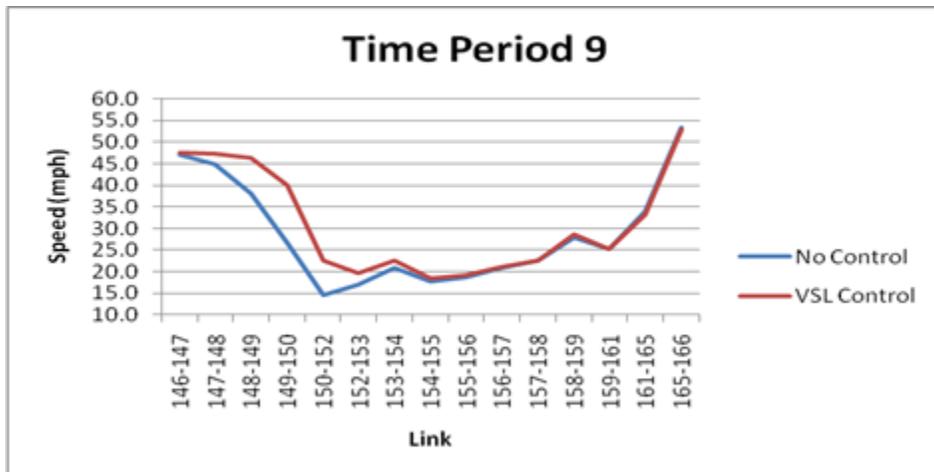
Figure 6-7. Continued



G

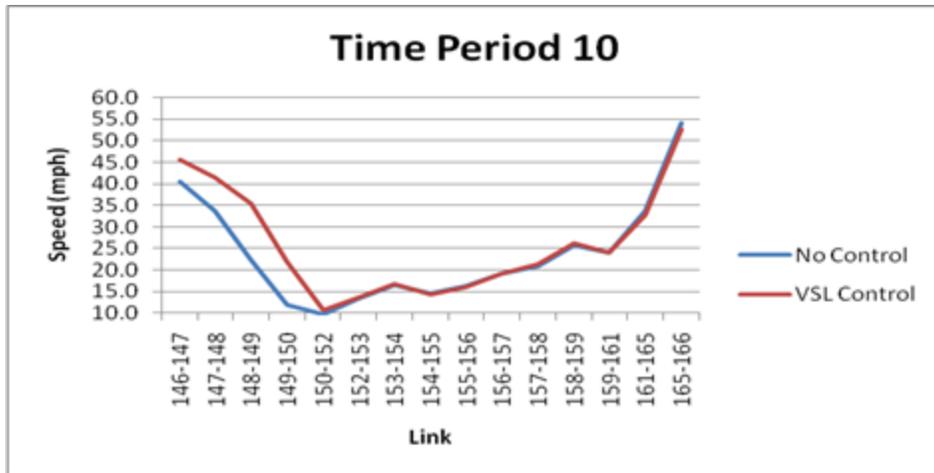


H

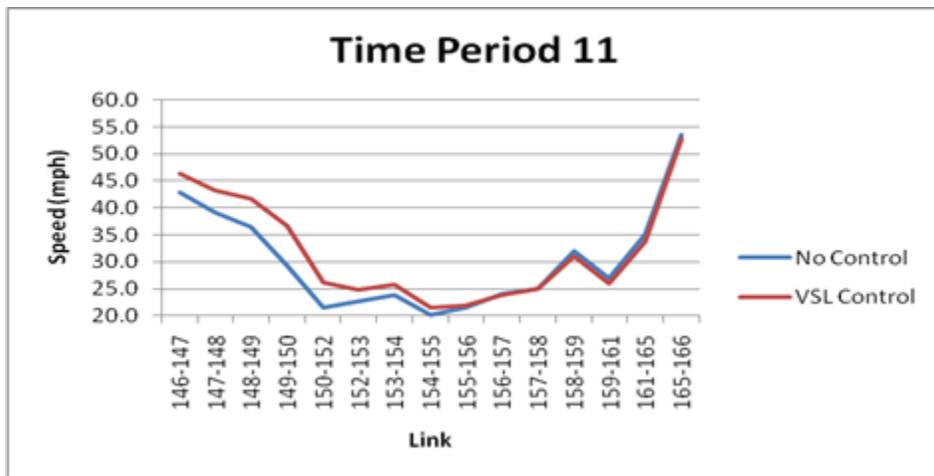


I

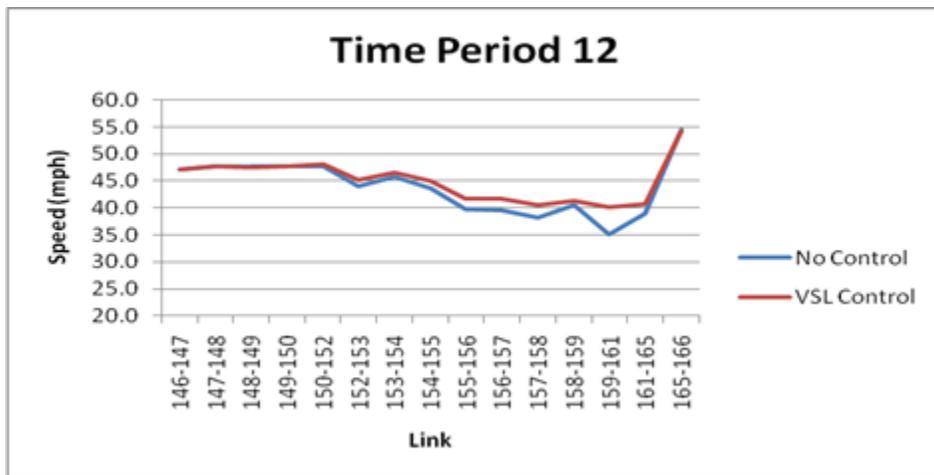
Figure 6-7. Continued



J

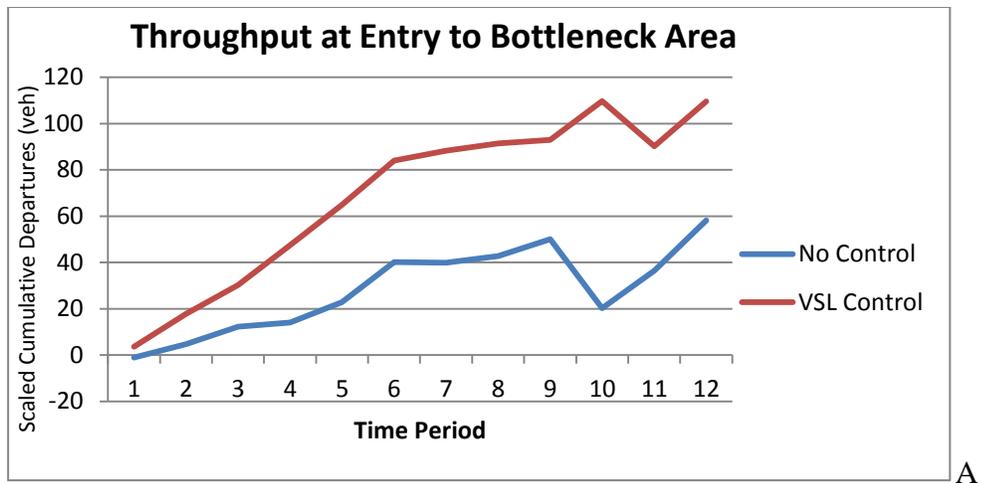


K

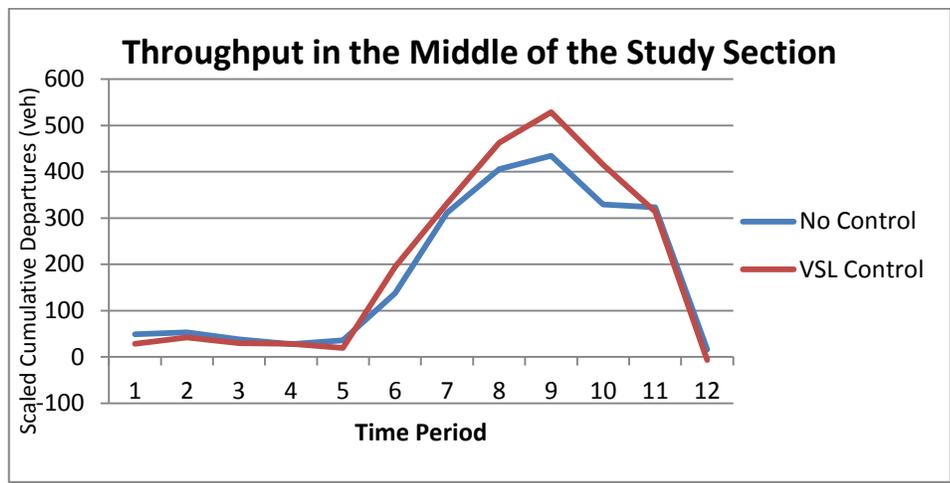


L

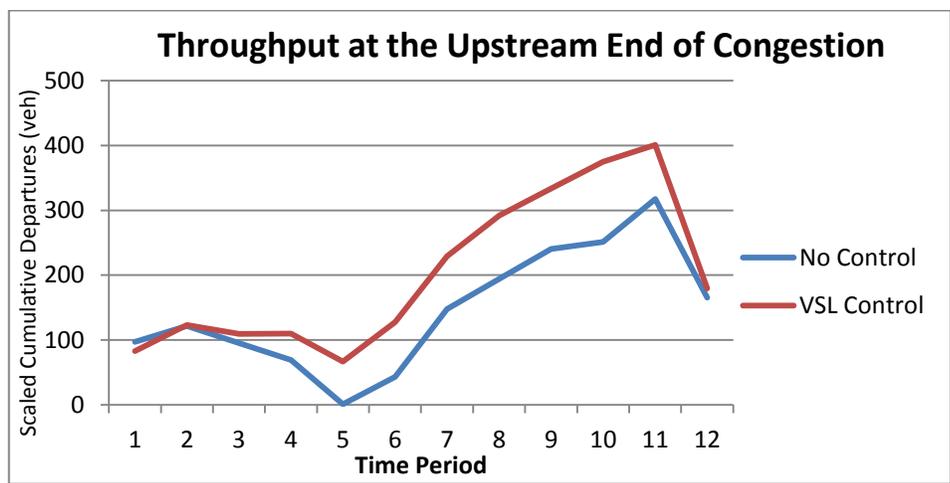
Figure 6-7. Continued



A



B



C

Figure 6-8. Throughput for the volume-based algorithm using one sign spaced 1 mile from the bottleneck source (threshold scenario 1) compared to the no-control scenario A) At entry to the bottleneck area (link 146-147). B) At the middle of the study section (link 153-154). C) At the upstream end of congestion (link 159-161).

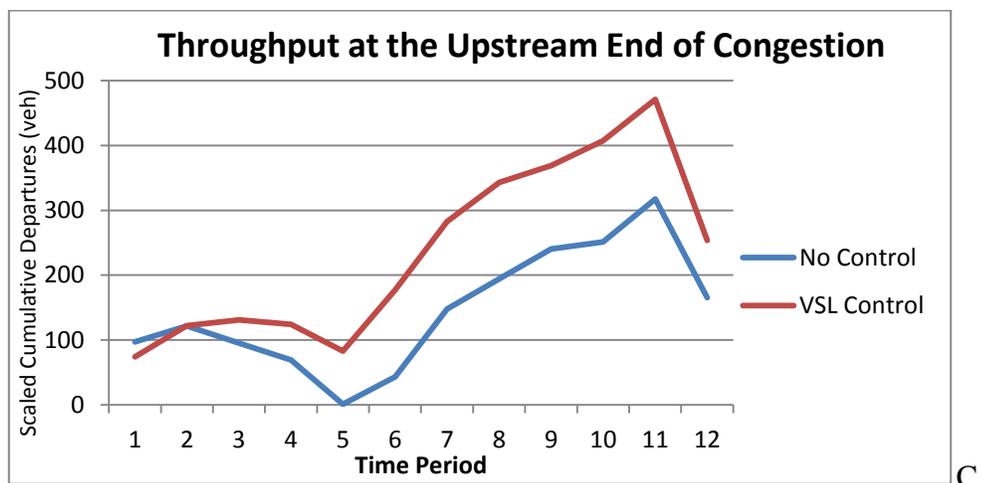
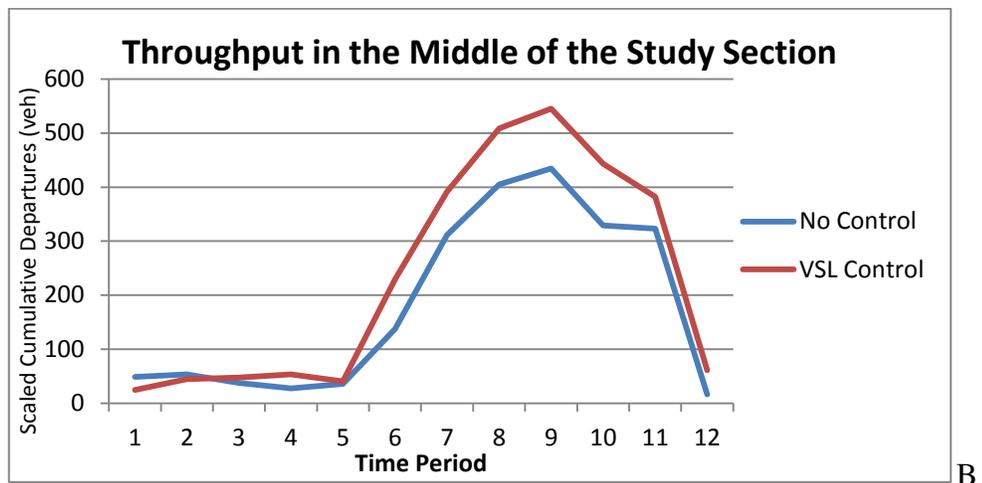
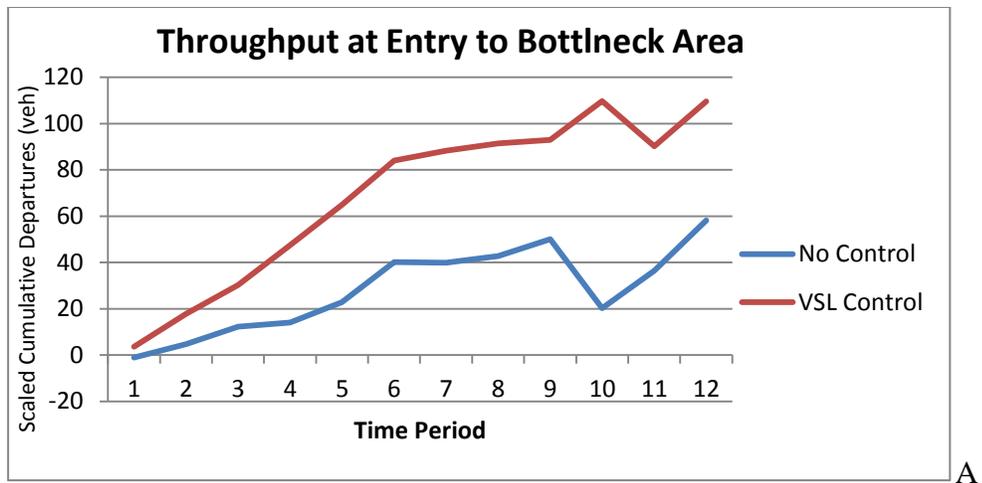


Figure 6-9. Throughput for the volume-based algorithm using two signs spaced 1/2 mile apart (threshold scenario 1) compared to the no-control scenario A) At entry to the bottleneck area (link 146-147). B) At the middle of the study section (link 153-154). C) At the upstream end of congestion (link 159-161).

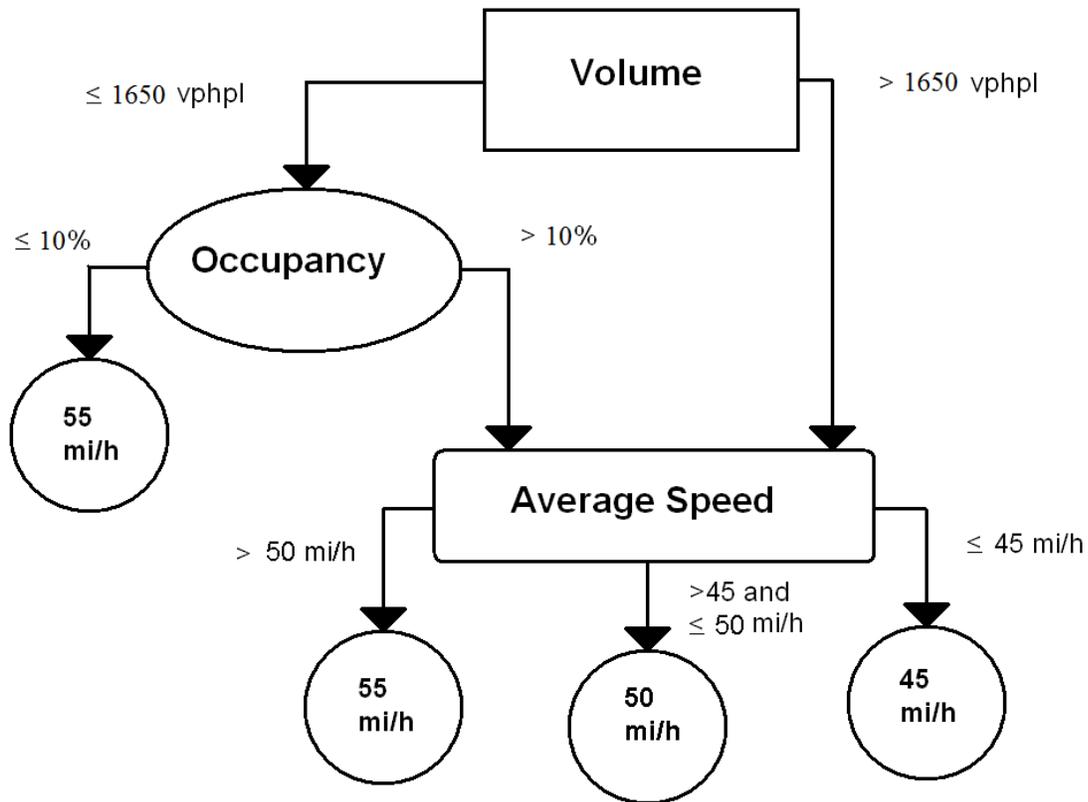


Figure 6-10. Logic tree thresholds used for the multiple parameter algorithm

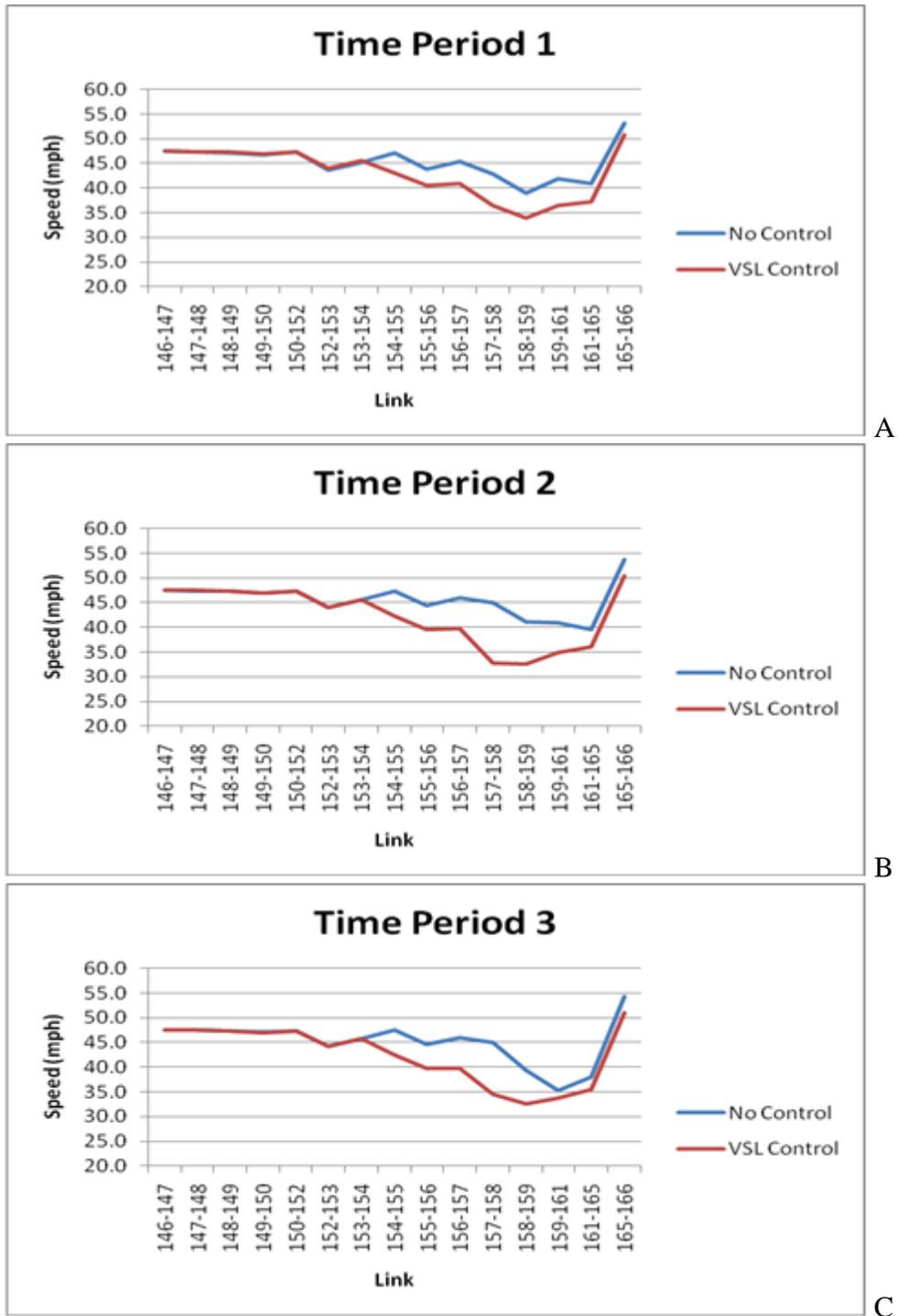
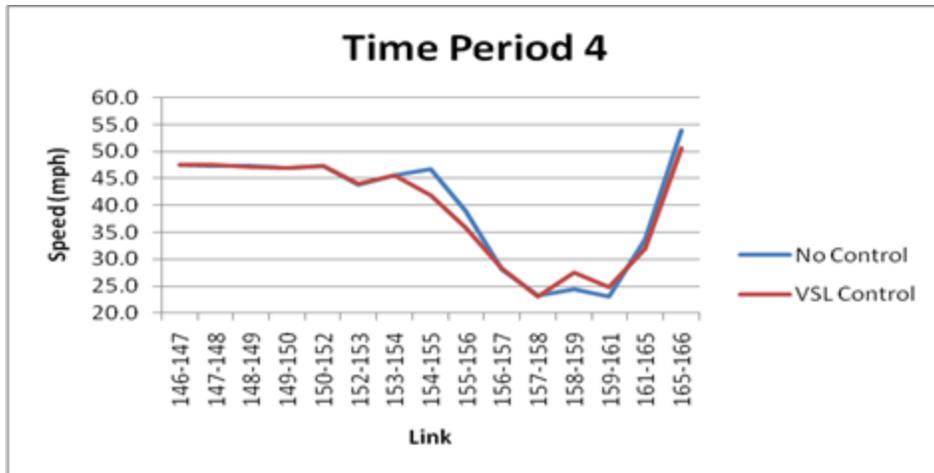
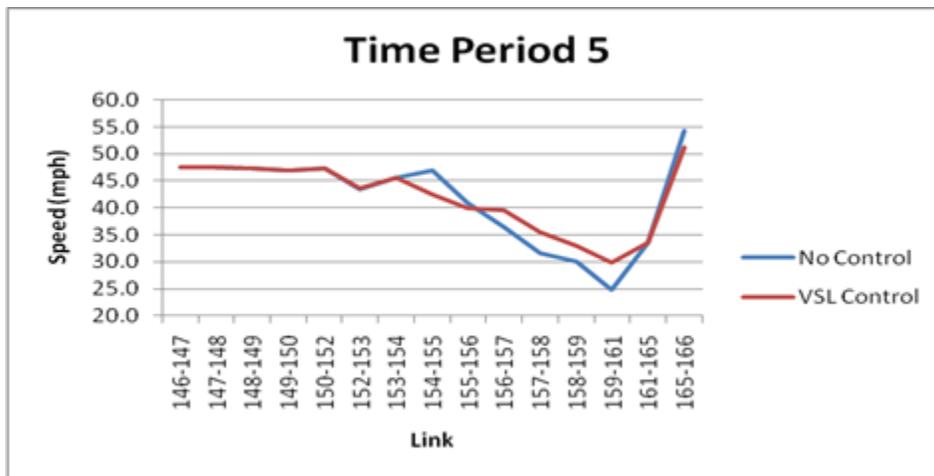


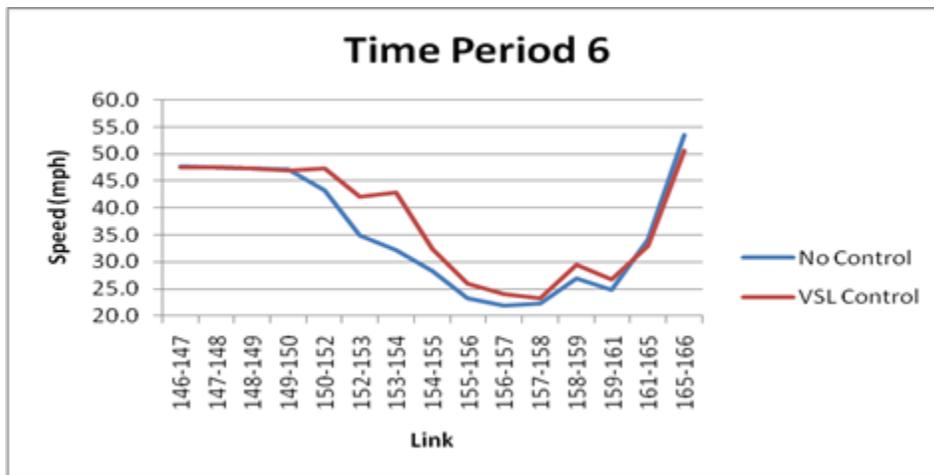
Figure 6-11. Speed profile for the multiple parameter--based algorithm using two signs spaced 1/2 mile apart compared to the no-control scenario A) Time period 1. B) Time Period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time Period 6. G) Time Period 7. H) Time Period 8. I) Time Period 9. J) Time Period 10. K) Time Period 11. L) Time Period 12.



D

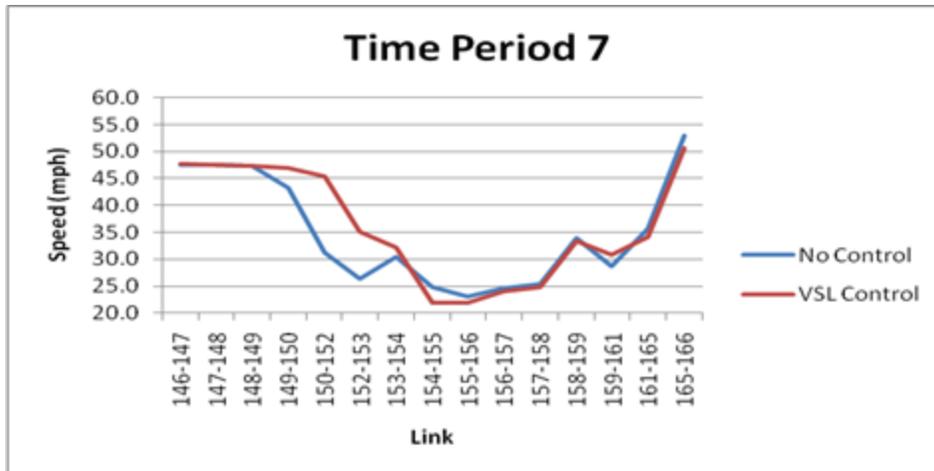


E

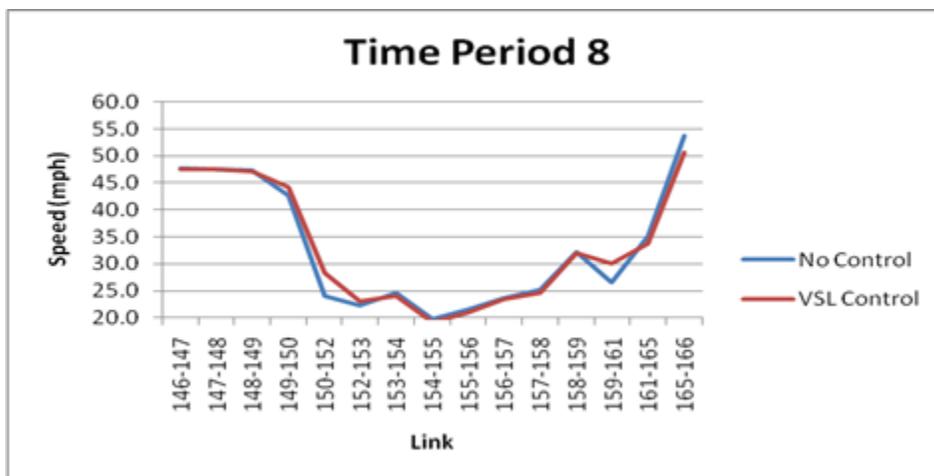


F

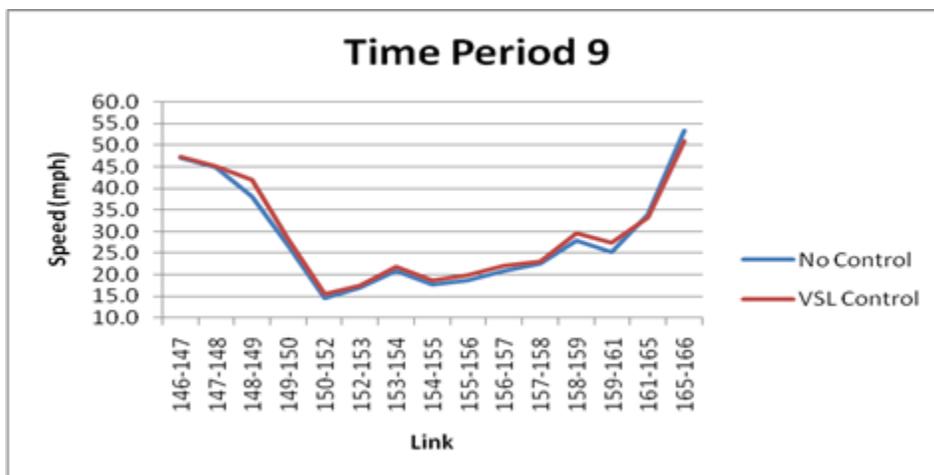
Figure 6-11. Continued



G

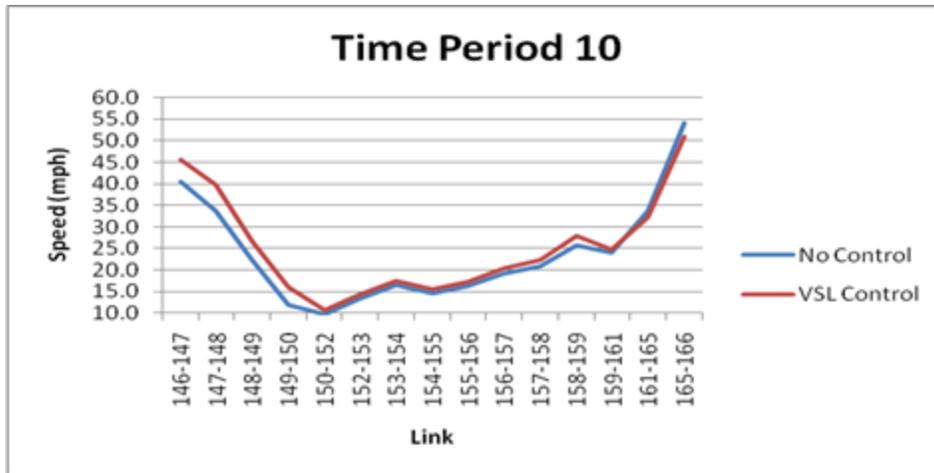


H

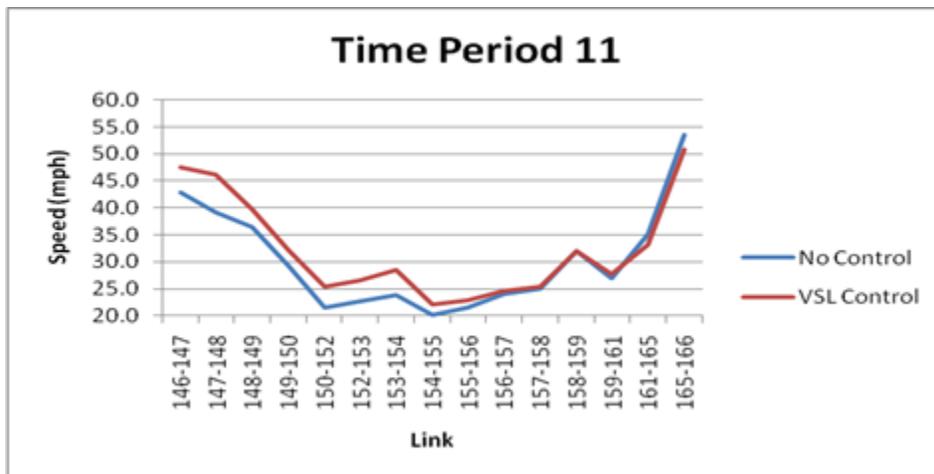


I

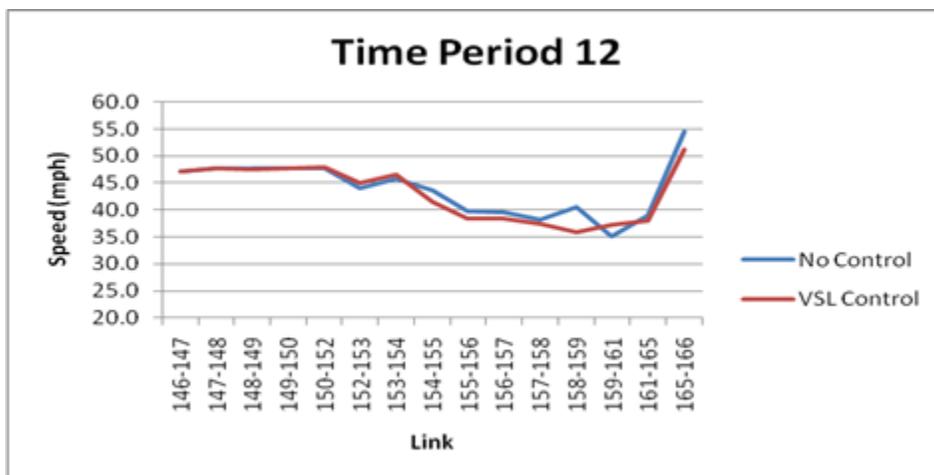
Figure 6-11. Continued



J

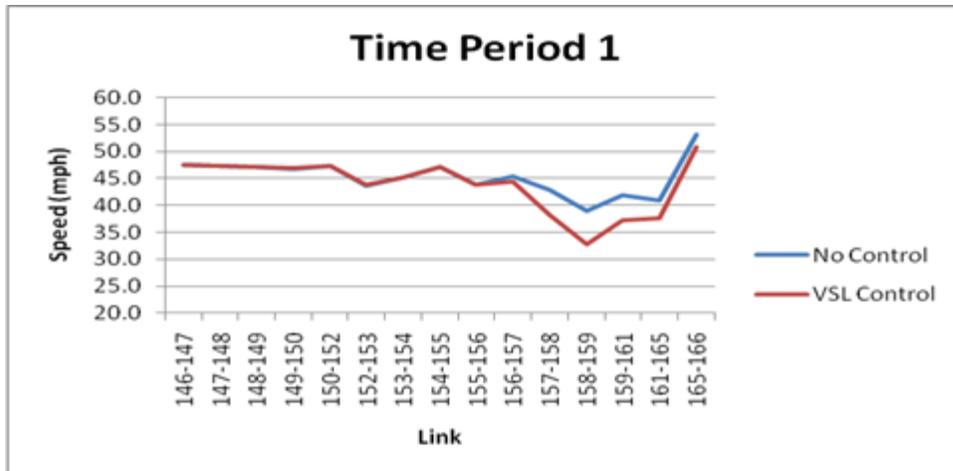


K

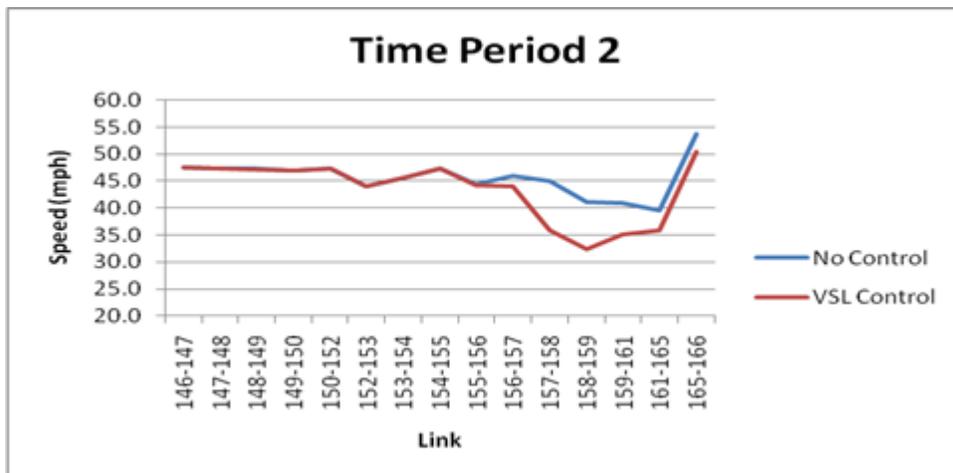


L

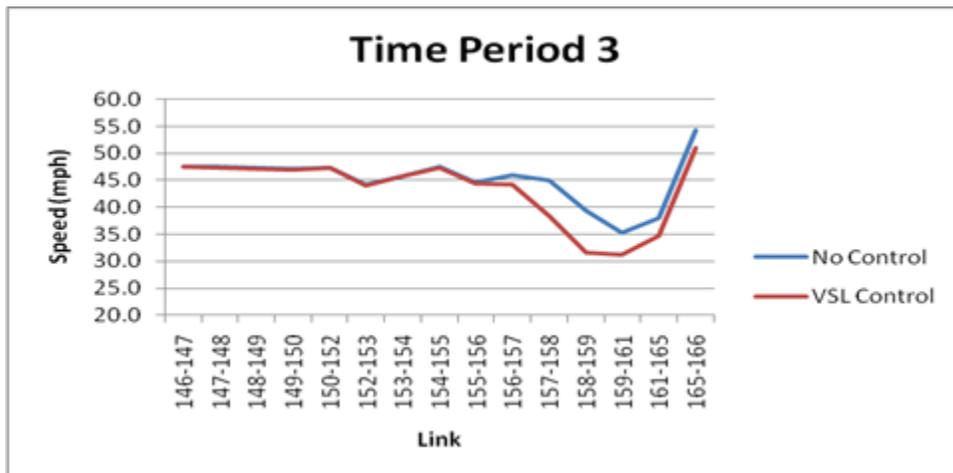
Figure 6-11. Continued



A

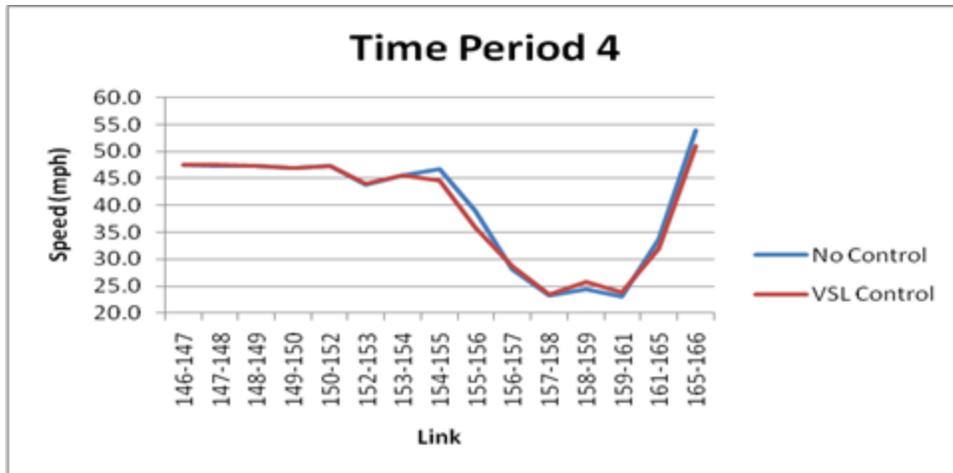


B

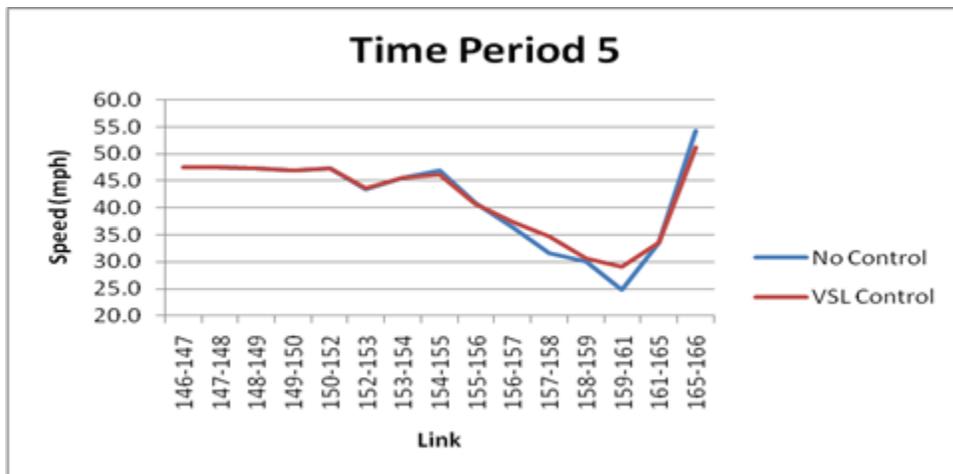


C

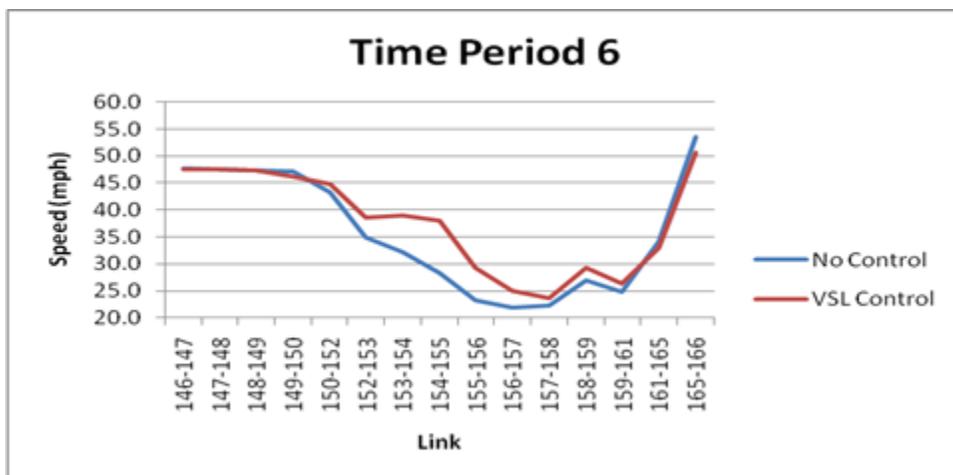
Figure 6-12. Speed profile for the multiple parameter-based algorithm using one sign spaced 1/2 mile from the bottleneck source compared to the no-control scenario A) Time period 1. B) Time Period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time Period 6. G) Time Period 7. H) Time Period 8. I) Time Period 9. J) Time Period 10. K) Time Period 11. L) Time Period 12.



D

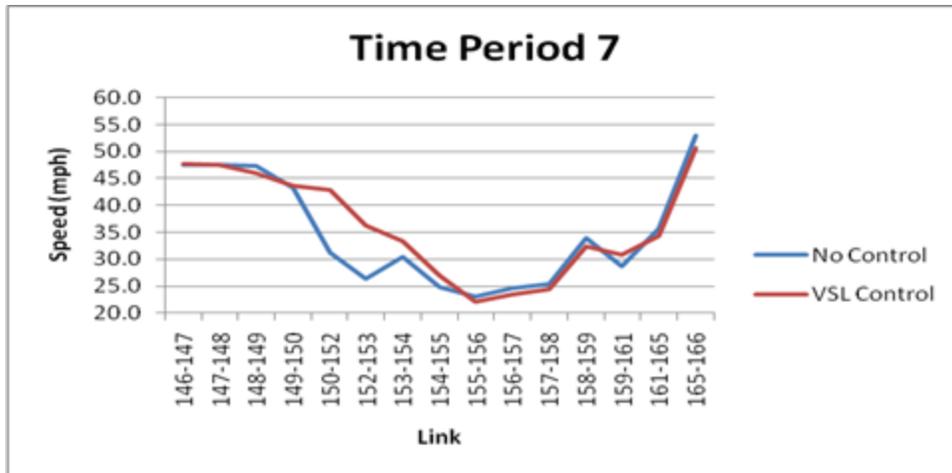


E

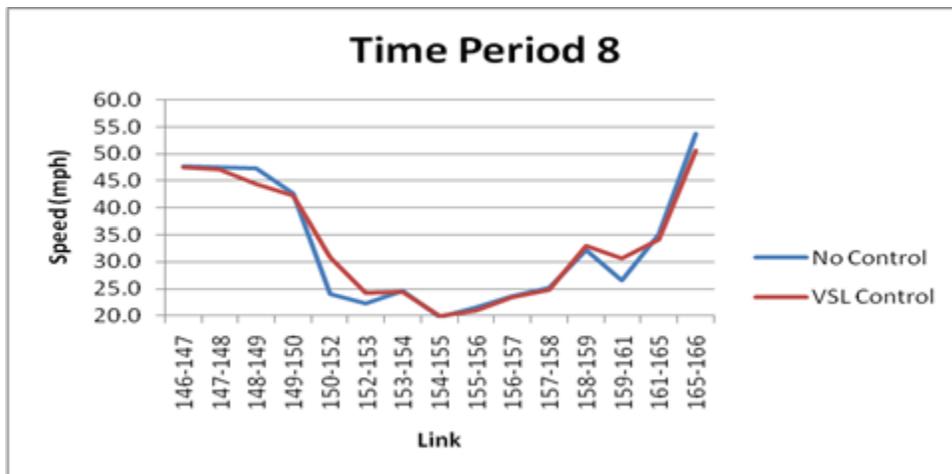


F

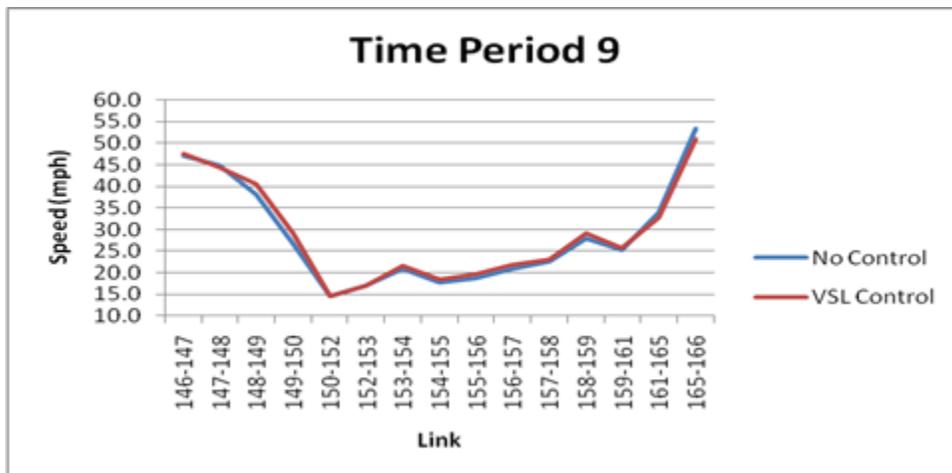
Figure 6-12. Continued



G

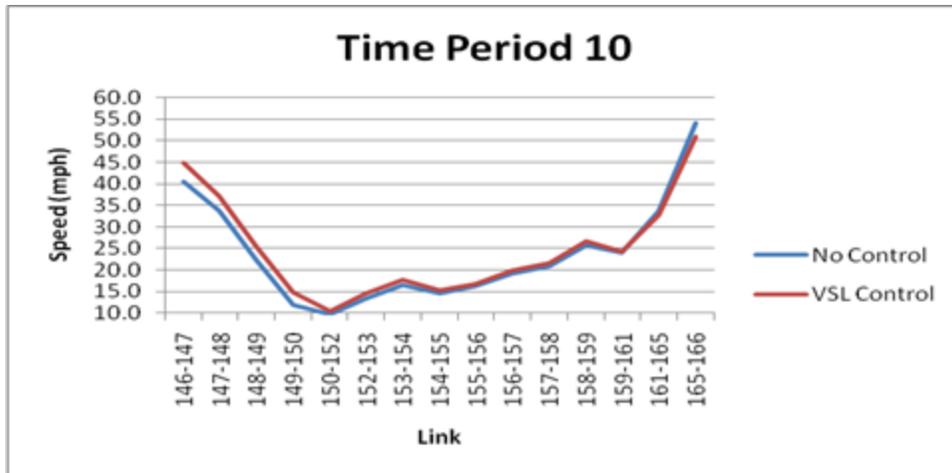


H

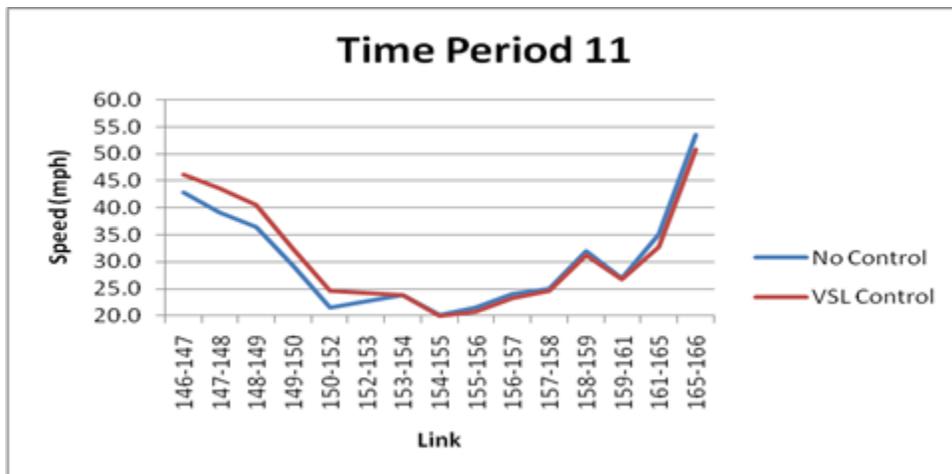


I

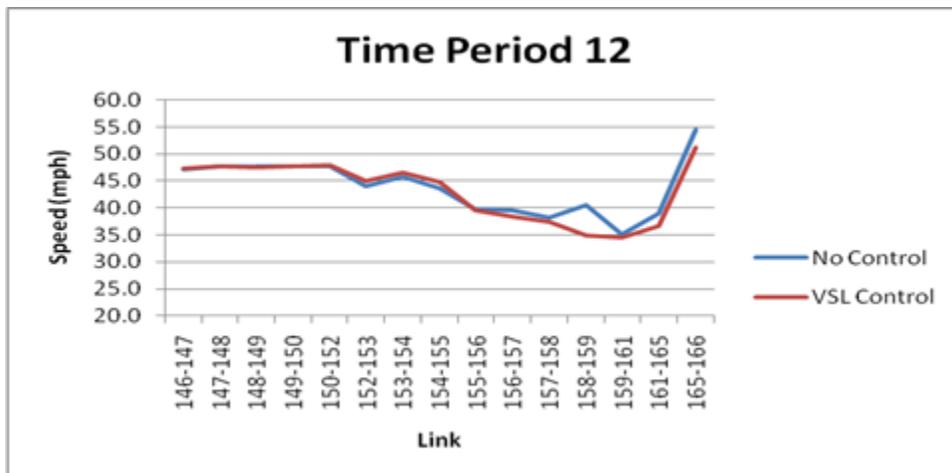
Figure 6-12. Continued



J

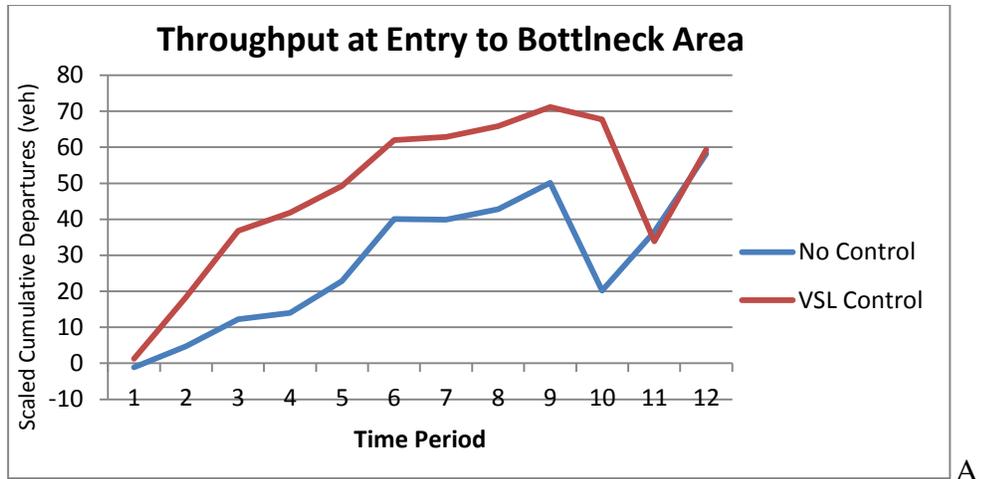


K

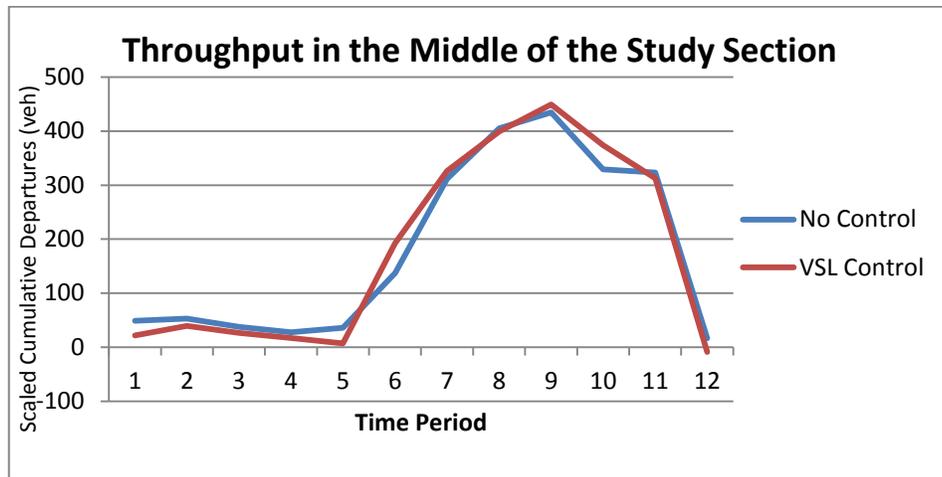


L

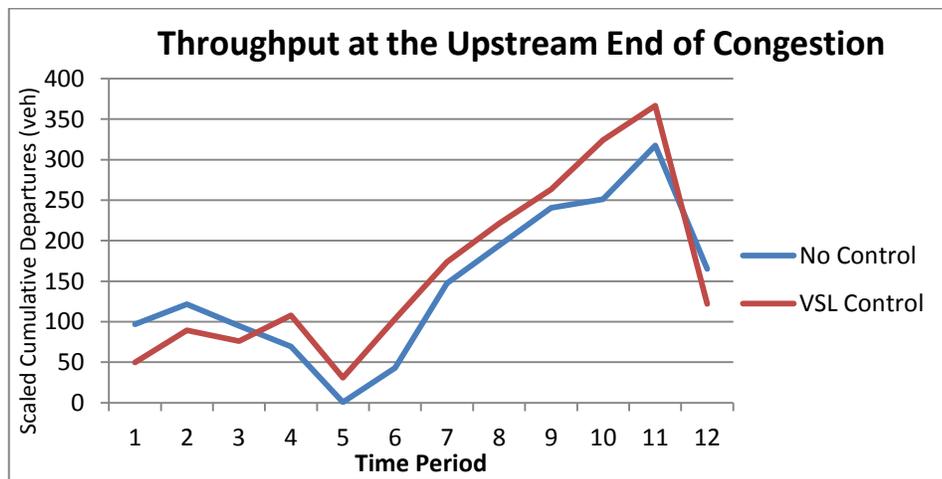
Figure 6-12. Continued



A



B



C

Figure 6-13. Throughput for the multiple-parameter-based algorithm using two signs spaced 1/2 mile apart compared to the no-control scenario A) At entry to bottleneck area (link 146-147). B) At the middle of the study section (link 153-154). C) At the upstream end of congestion (link 159-161).

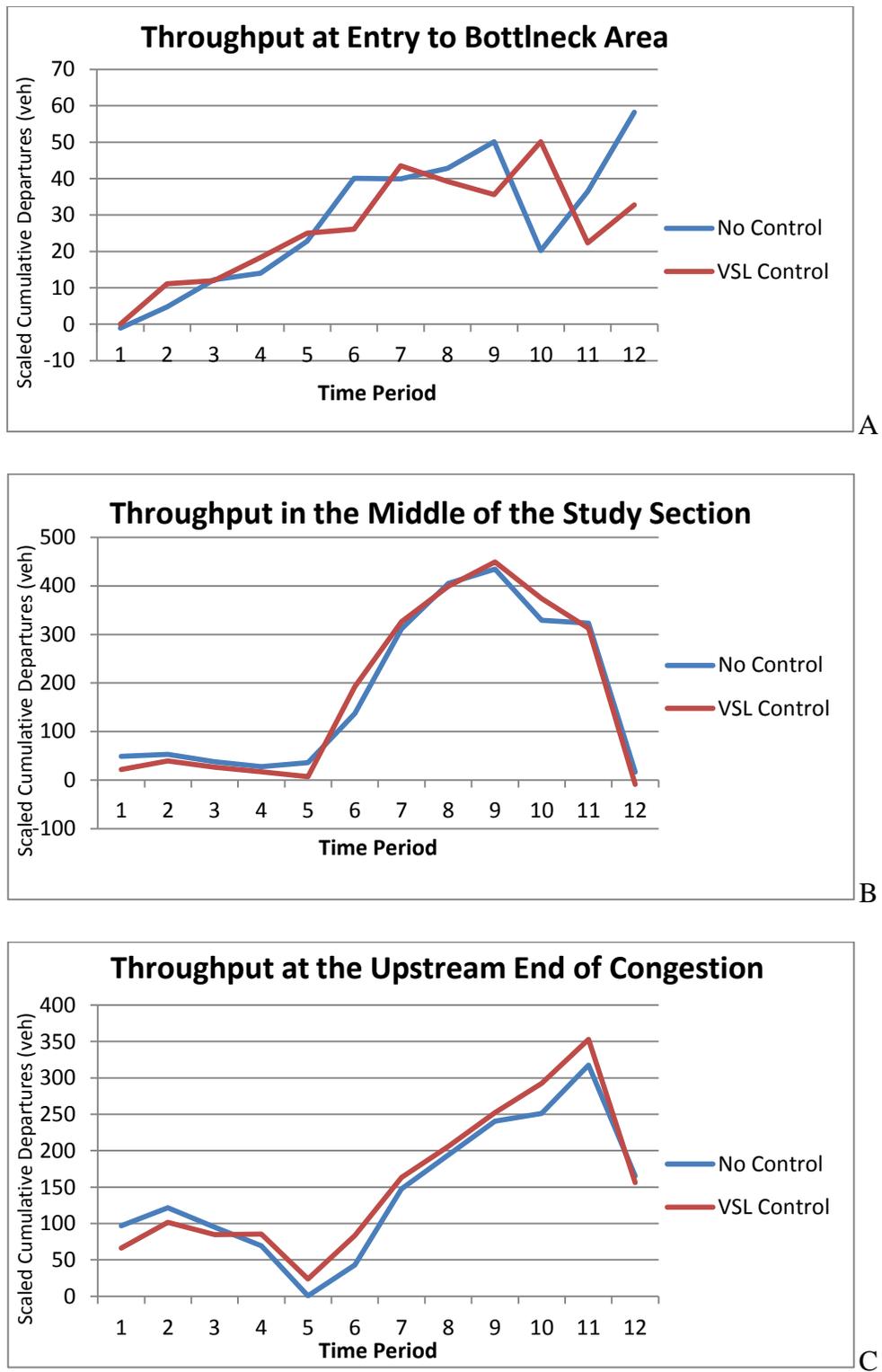
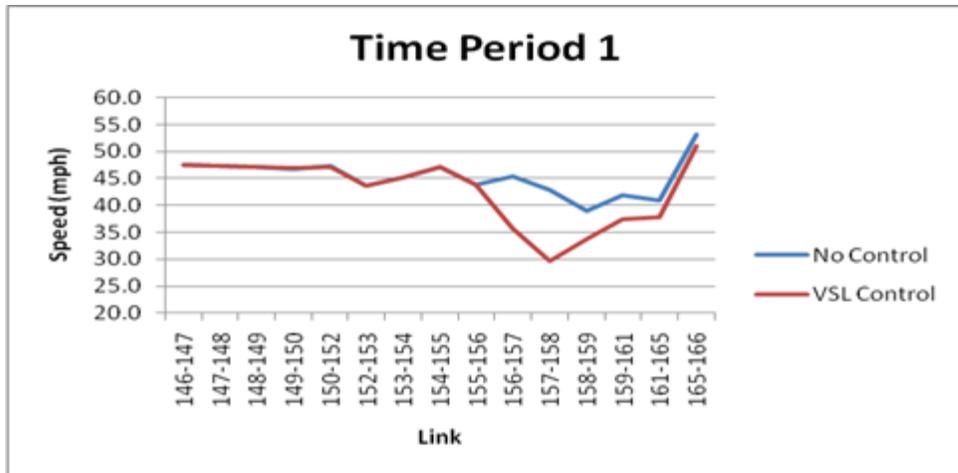
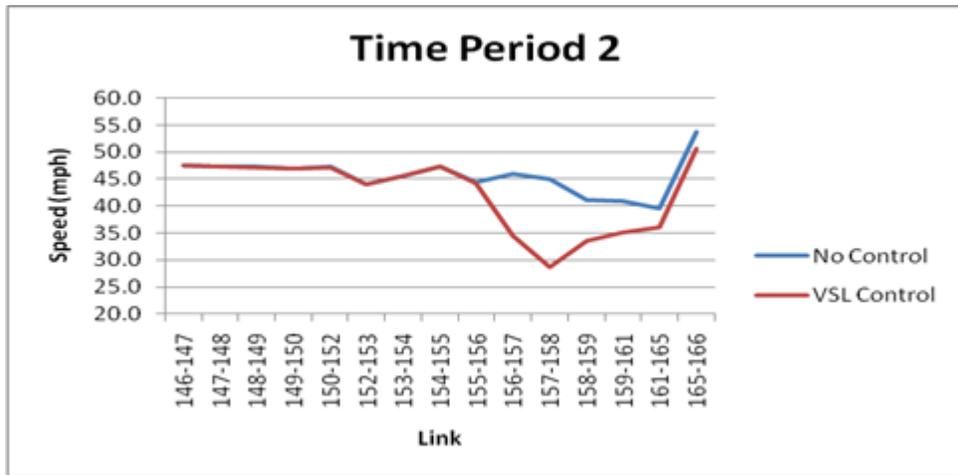


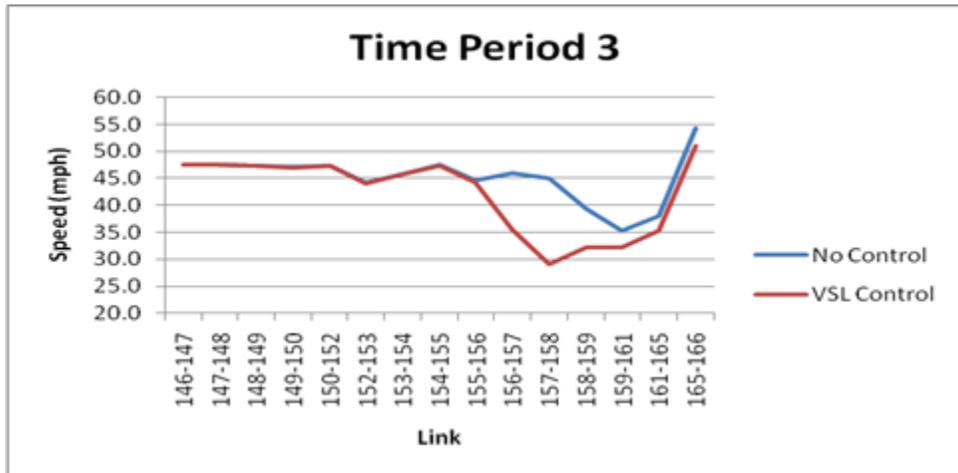
Figure 6-14. Throughput for the multiple-parameter-based algorithm using one sign spaced 1/2 mile from the bottleneck source compared to the no-control scenario A) At entry to the bottleneck area (link 146-147). B) At the middle of the study section (link 153-154). C) At the upstream end of congestion (link159-161).



A

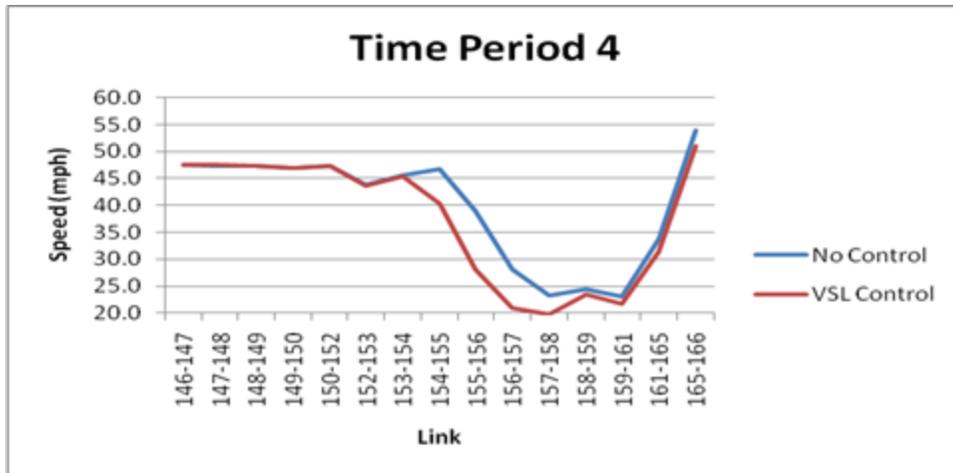


B

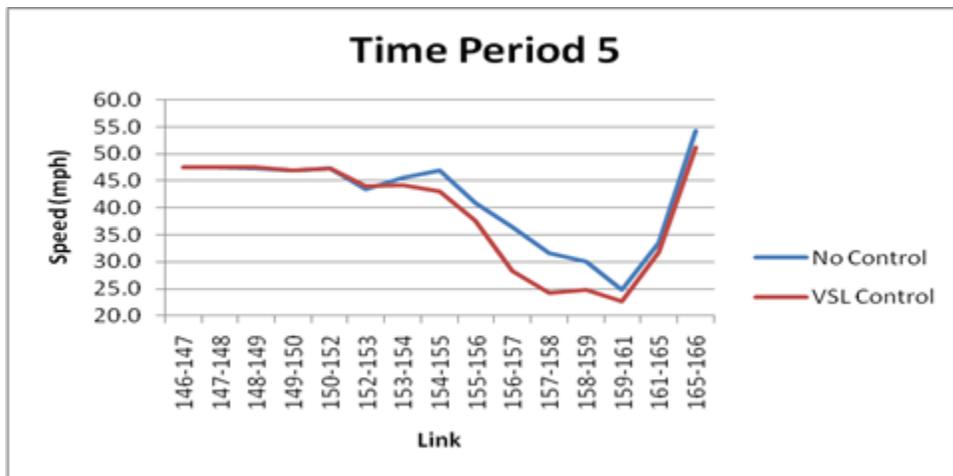


C

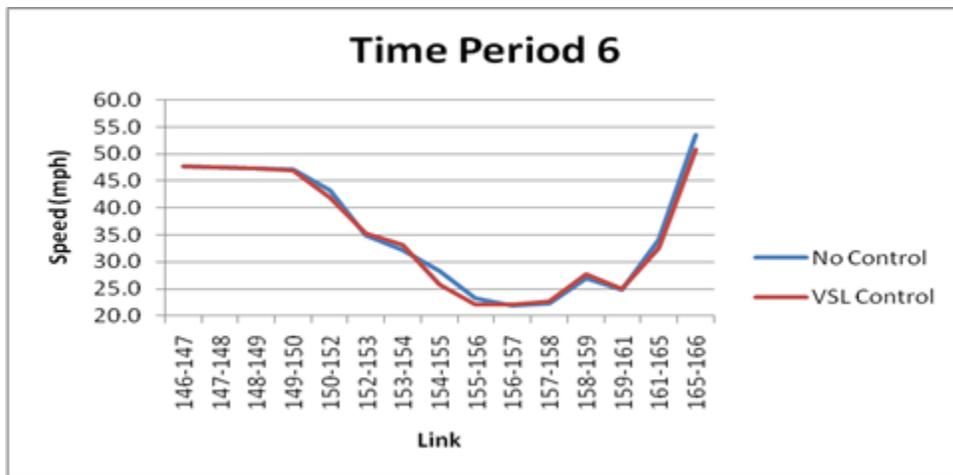
Figure 6-15. Speed profile for the multiple parameter-based algorithm using one sign spaced 1 mile from the bottleneck source compared to the no-control scenario A) Time period 1. B) Time Period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time Period 6. G) Time Period 7. H) Time Period 8. I) Time Period 9. J) Time Period 10. K) Time Period 11. L) Time Period 12.



D

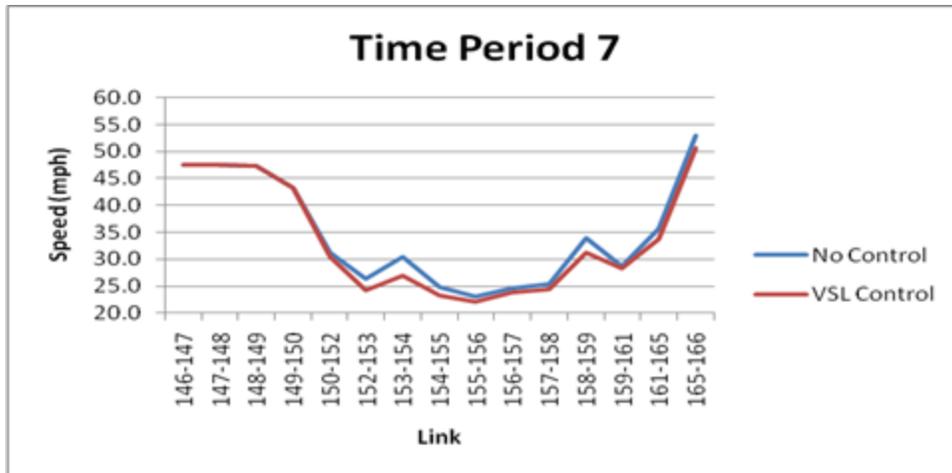


E

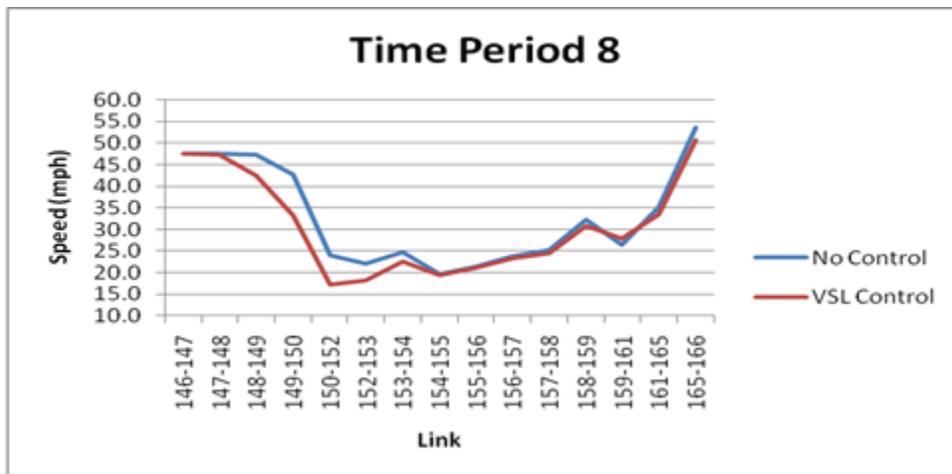


F

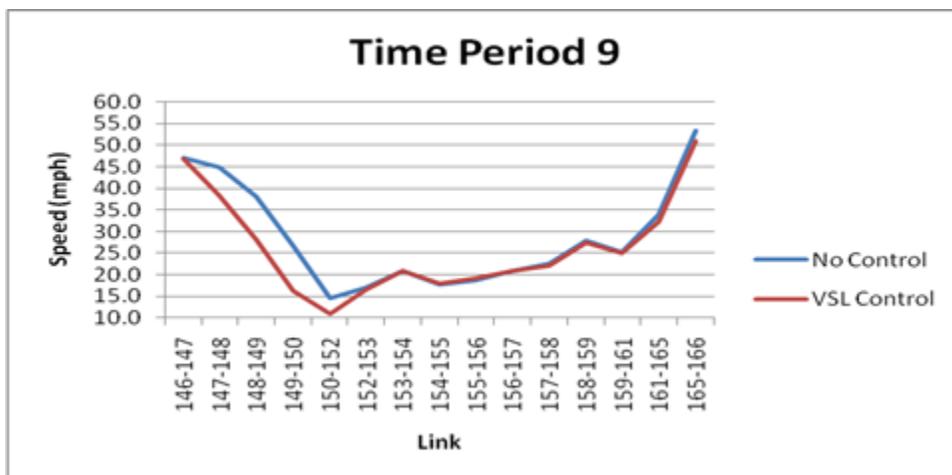
Figure 6-15. Continued



G

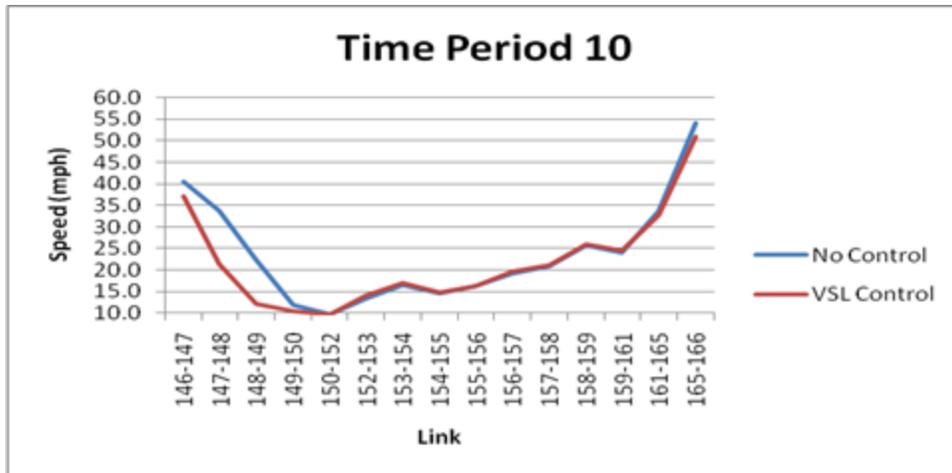


H

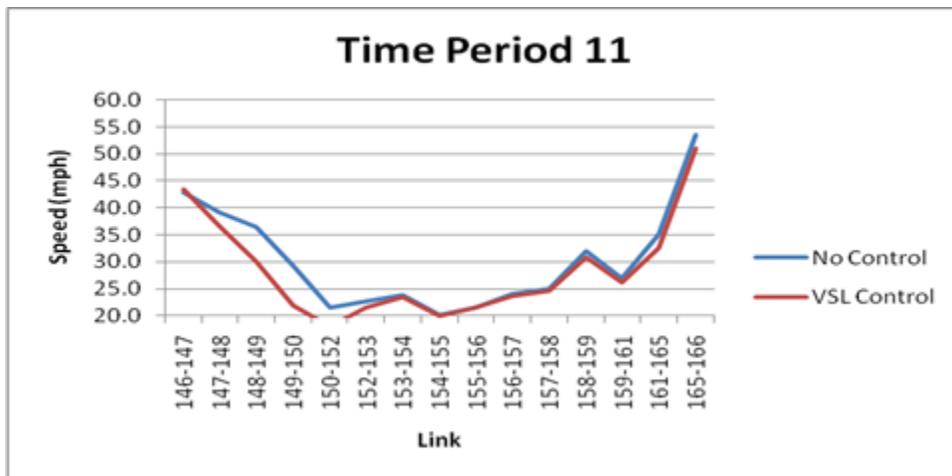


I

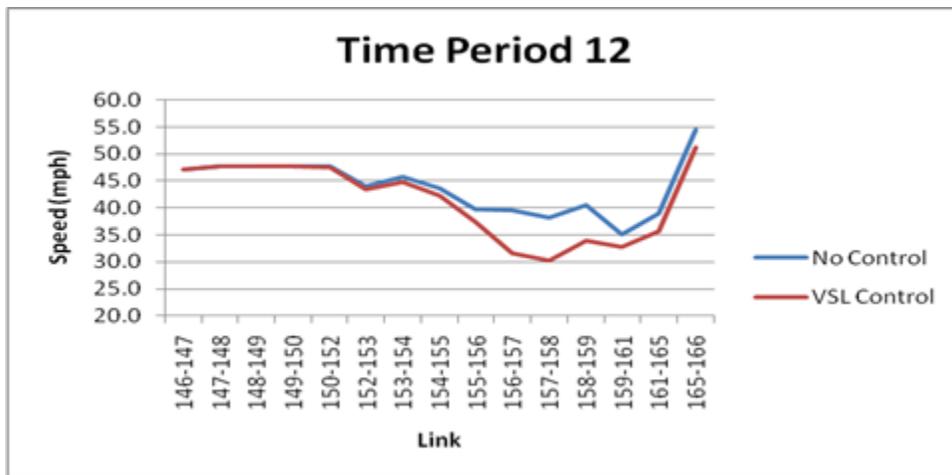
Figure 6-15. Continued



J

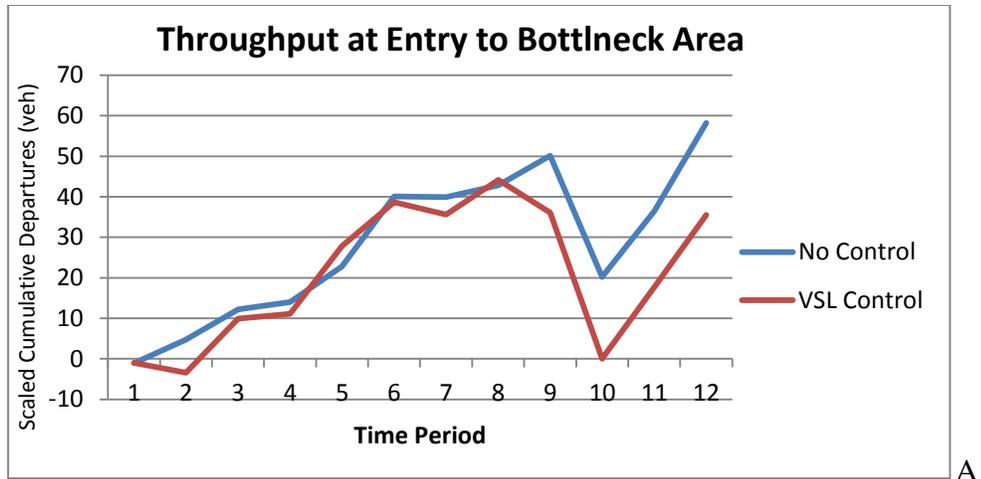


K

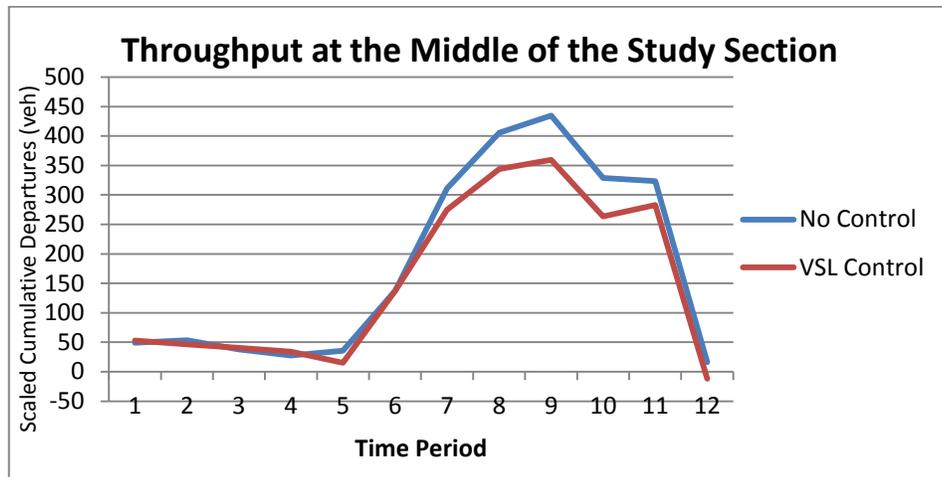


L

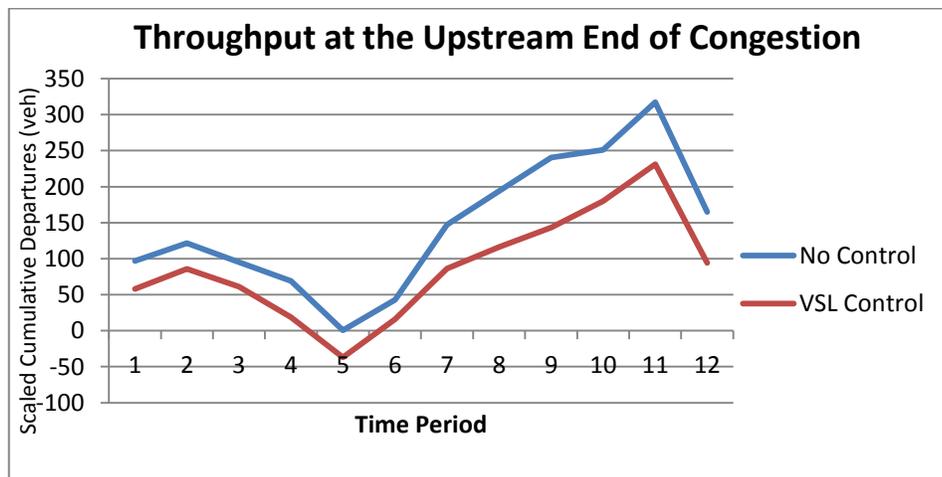
Figure 6-15. Continued



A



B



C

Figure 6-16. Throughput for the multiple-parameter-based algorithm using one sign spaced 1 mile from the bottleneck source compared to the no-control scenario A) At entry to bottleneck area (link 146-147). B) At the middle of the study section (link 153-154). C) At the upstream end of congestion (link 159-161).

CHAPTER 7 SUMMARY AND CONCLUSIONS

Variable speed limits (VSLs) have been used upstream of bottlenecks with recurring congestion as a way to dampen the shockwave produced once congestion starts. However, the exact effects of VSL on traffic flow are not well understood, and the literature provides conflicting conclusions with respect to the effectiveness of VSL installations in increasing overall speeds and throughput. Simulation is a very effective tool in evaluating alternatives under completely controlled conditions which cannot be achieved in the field. It is also very effective in providing a comprehensive picture of traffic operations in time and space.

This study creates a test-bed for evaluating different VSL control algorithms and implementations through the use of the micro-simulator Corridor Simulation (CORSIM). A run time extension (RTE) tool is built and used to simulate VSL along a freeway section, to allow the evaluation of multiple algorithms under varying conditions.

The RTE is created to model the implementation of a VSL system using C++ code. The dynamic link library (DLL) created interfaces with the CORSIM simulation at specified call points. Detector data are averaged every 1 minute and the speed limit is evaluated based on given thresholds. Three different algorithms are tested along with a number of threshold scenarios, and sign positioning. One algorithm uses thresholds based on occupancy, the second one uses thresholds based on volumes, and the last one uses thresholds based on volume, occupancy, and average travel speed.

The following were concluded:

- Through all 24 of the scenarios tested, all but one show an improvement in terms of average travel speed and total travel time. The magnitude of the change is relatively small when averaged through the entire network.
- The throughput was found to increase for most of the VSL scenarios tested by a maximum of 30 to 90 vehicles over a given 15 minute time period.

- Of the scenarios tested, the volume-based algorithm displays the largest improvement in terms of increased average speed, and reduced travel time.
- The effect of the VSL may not be immediately seen if one examines conditions only at the bottleneck. The area upstream of the bottleneck shows much greater traffic improvements than the bottleneck itself.
- Improper selection of thresholds or sign positioning can cause traffic conditions to become worse than if no VSL control is used. This is shown by the multiple parameter algorithm (scenario 2), where conditions worsened for every evaluation parameter and time period.
- There was no consistent trend in traffic conditions as a function of the number and location of speed limit signs. The best sign positioning was found to be highly dependent on the type of algorithm and specific thresholds selected.
- The results and conclusions made by this study assume a level of compliance from motorists. In order for these results to parallel real world implementation, there must be the same level of compliance. This emphasizes the need for enforcement of the speed limits when they are implemented in the field.

The inclusion of a user interface in the normal installation of CORSIM should be considered for future releases of the software. Recommendations for a permanent interface include the following:

- Develop an alternative to the rolling speed limit approach. This could be accomplished by assigning speed limit changes on a vehicle basis, rather than on a link basis.
- Include input screens for each type of VSL algorithm. It is recommended to have volume, occupancy, and speed-based VSL algorithms available as options.
- Allow the user to specify threshold values to be associated with speed limit changes for each type of algorithm simulated.
- Incorporate an interface to place VSL signs at a desired location, and link it to specified detectors using a visual interface.

Recommendations for future research include the following:

- This study did not attempt to thoroughly evaluate and compare the three VSL algorithms, nor to obtain optimal thresholds for each type of algorithm. An optimization-type study could be performed to obtain optimal thresholds, sign locations, and detector locations.

- The study could be extended to consider additional bottlenecks along the I-95 corridor.
- Research considering ramp metering in conjunction with VSL would be useful in determining the mechanism through which these two tools would interact, and developing guidelines for optimizing their joint operation.
- Field testing of the algorithms tested here and a comparison between simulation and field results would be useful in validating, and as necessary enhancing the simulation algorithm.

APPENDIX A
CALIBRATION RESULTS FOR I-95 NETWORK

This appendix provides the calibration results of the I-95 CORSIM simulation. The network was calibrated based on 11 detector locations. This appendix provides the volume calibration information over all 12 time periods, followed by the speed calibration data over all 12 time periods.

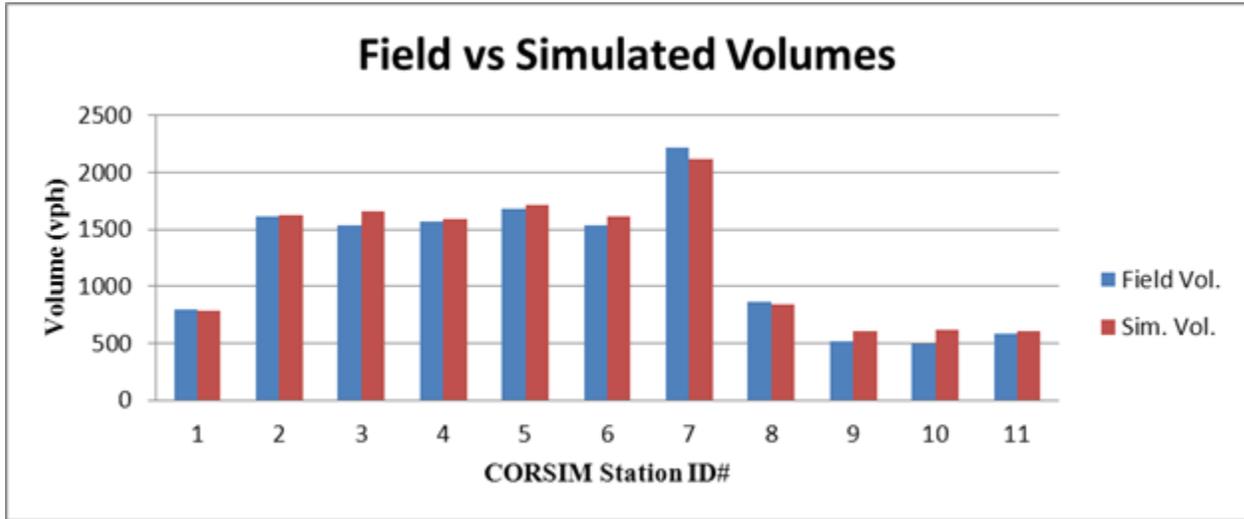


Figure A-1. Volume calibration for time period 1

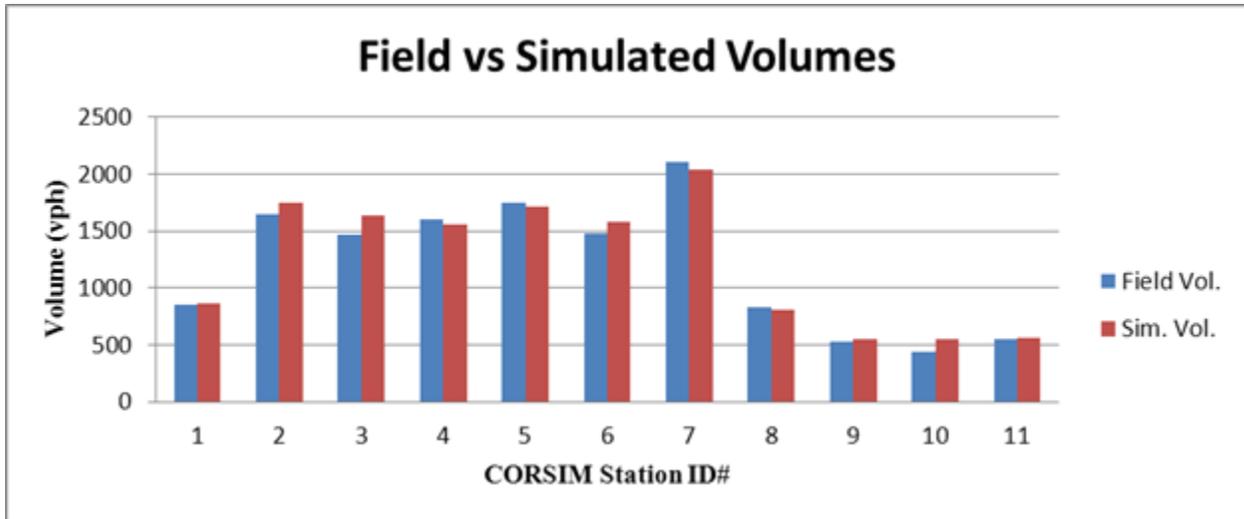


Figure A-2. Volume calibration for time period 2

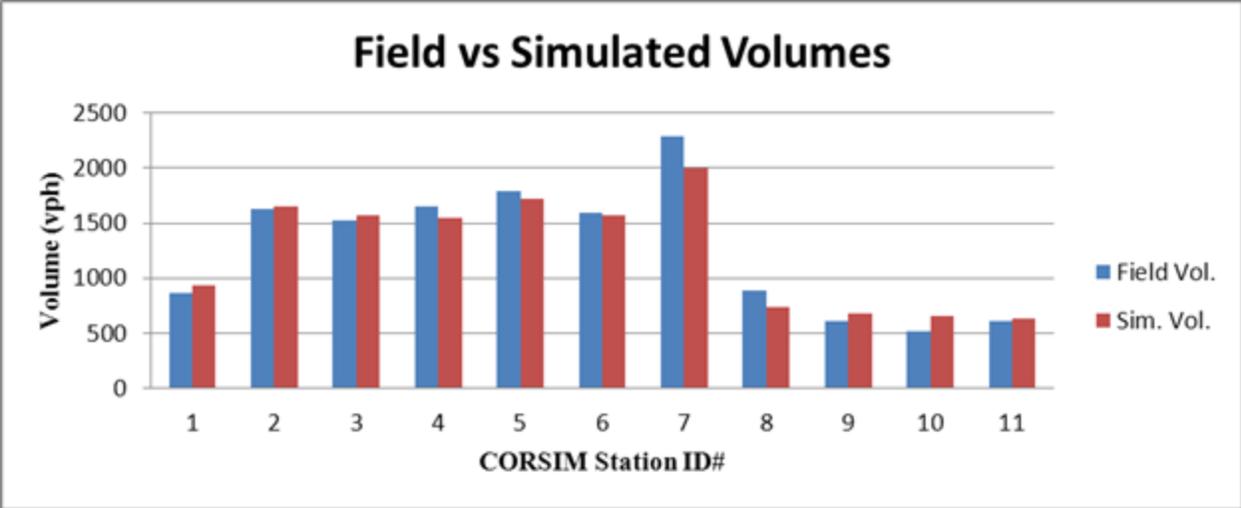


Figure A-3. Volume calibration for time period 3

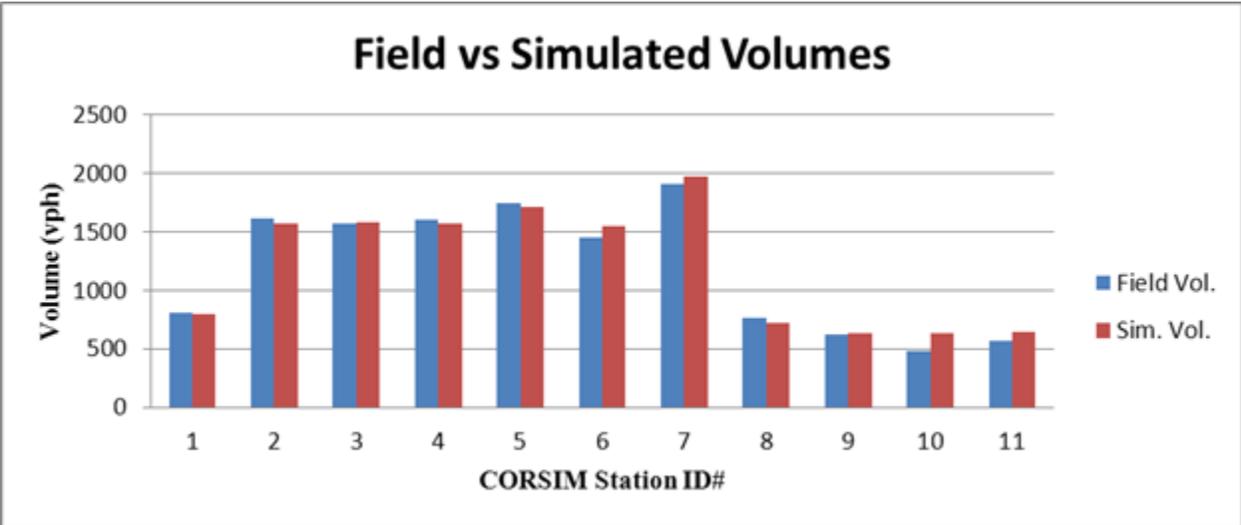


Figure A-4. Volume calibration for time period 4

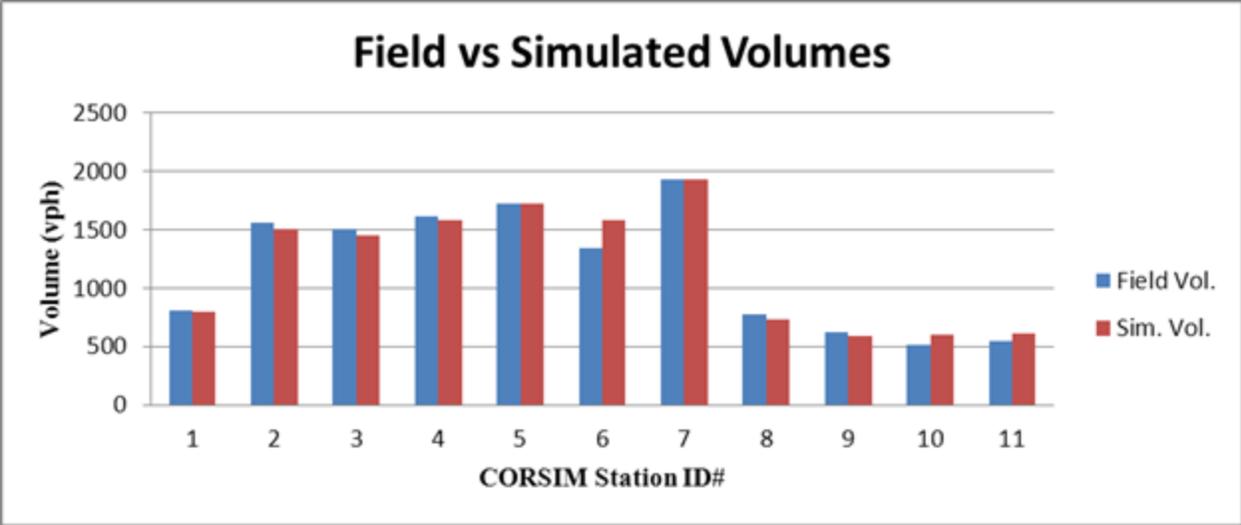


Figure A-5. Volume calibration for time period 5

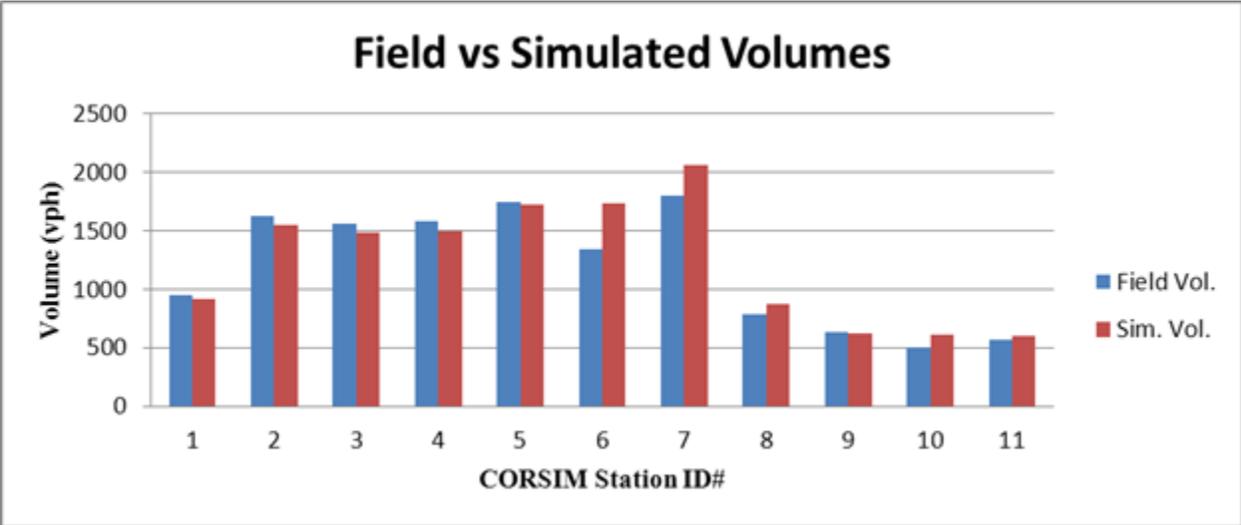


Figure A-6. Volume calibration for time period 6

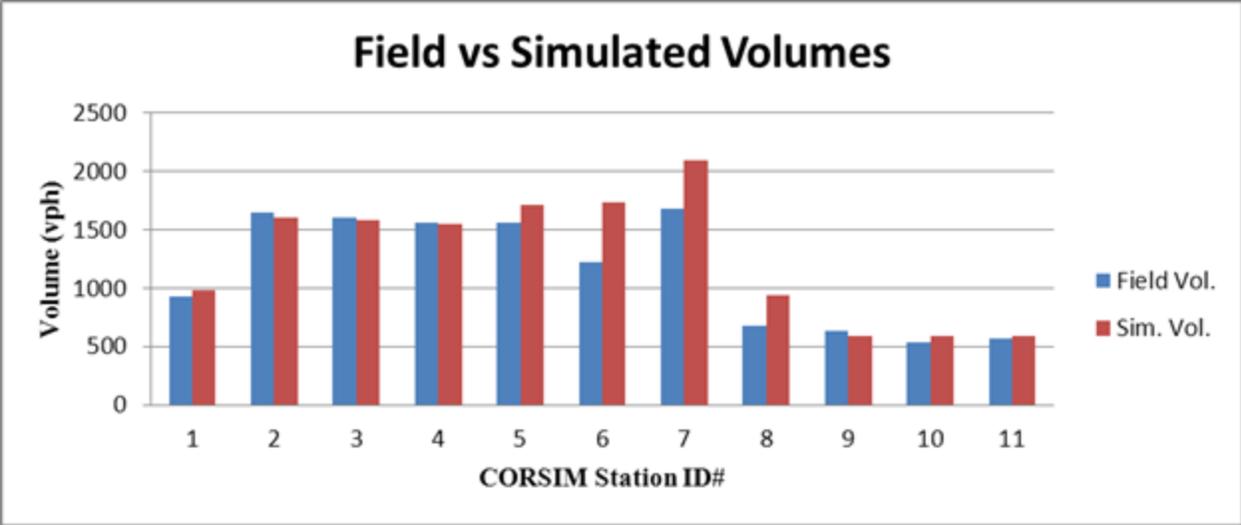


Figure A-7. Volume calibration for time period 7

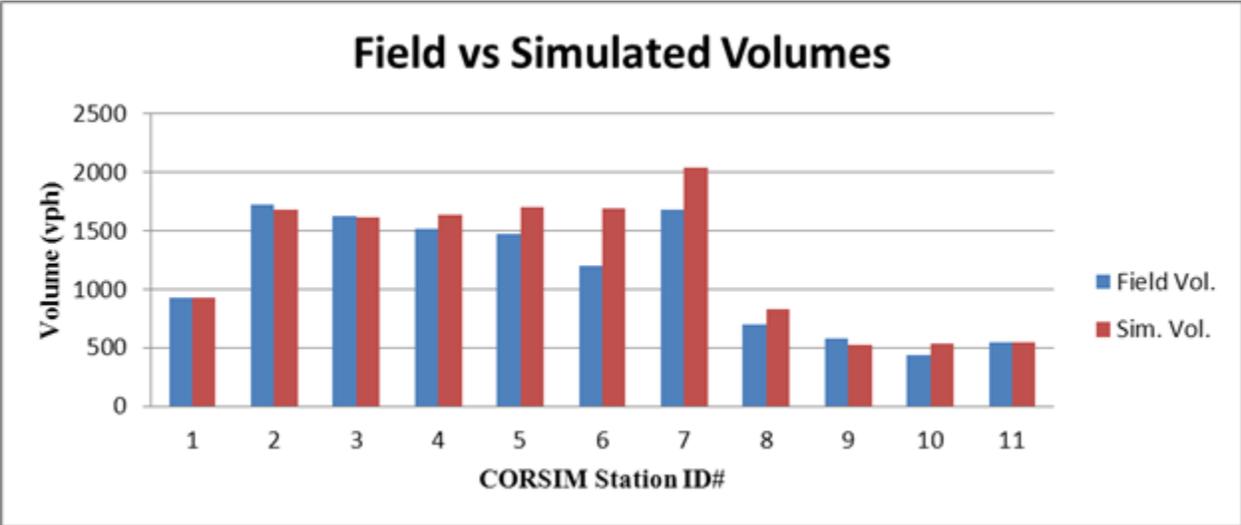


Figure A-8. Volume calibration for time period 8

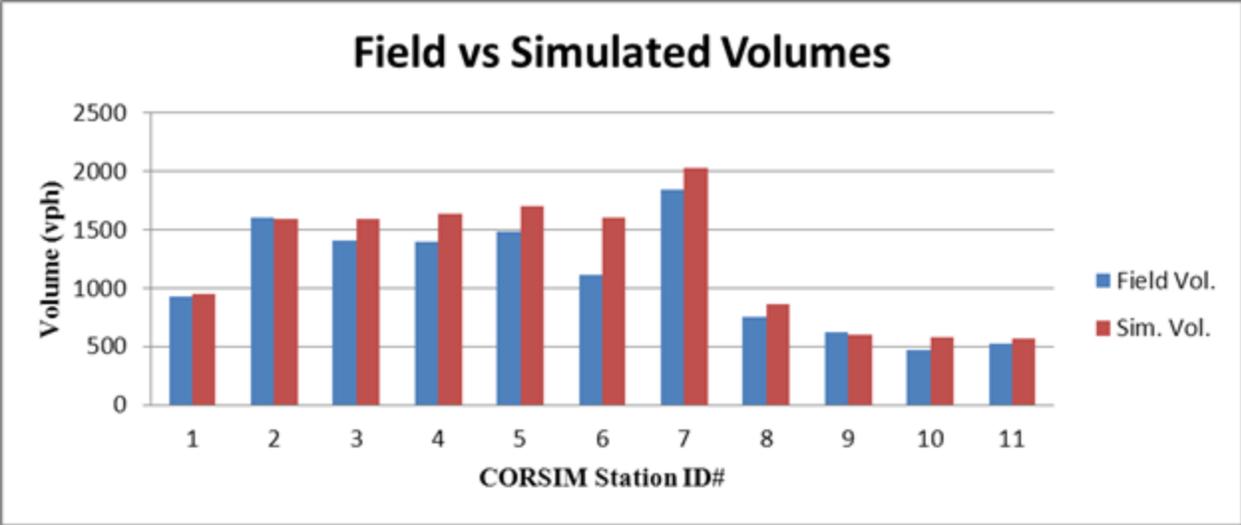


Figure A-9. Volume calibration for time period 9

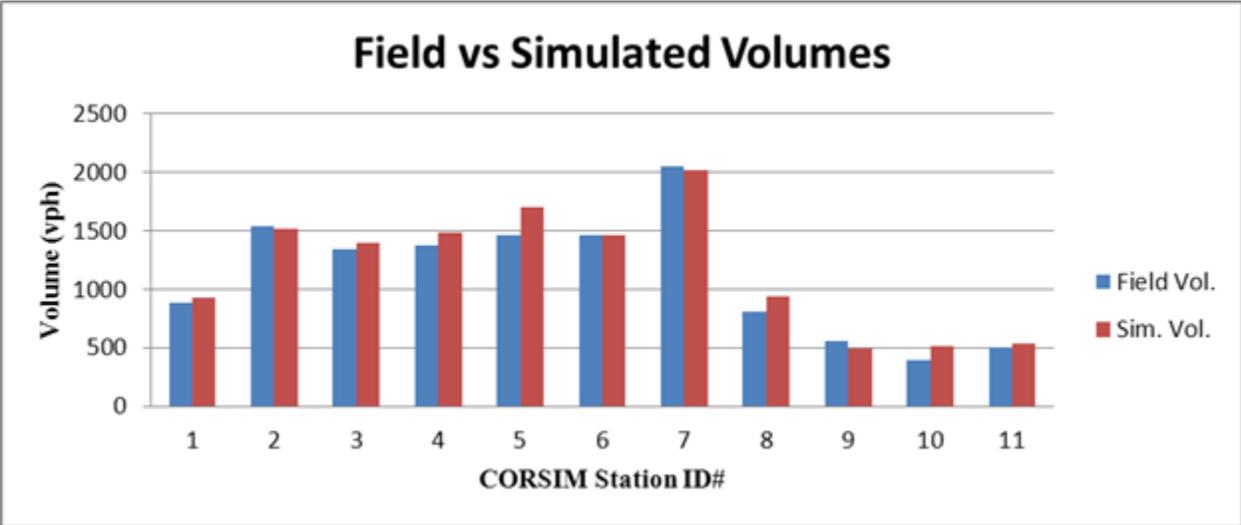


Figure A-10. Volume calibration for time period 10

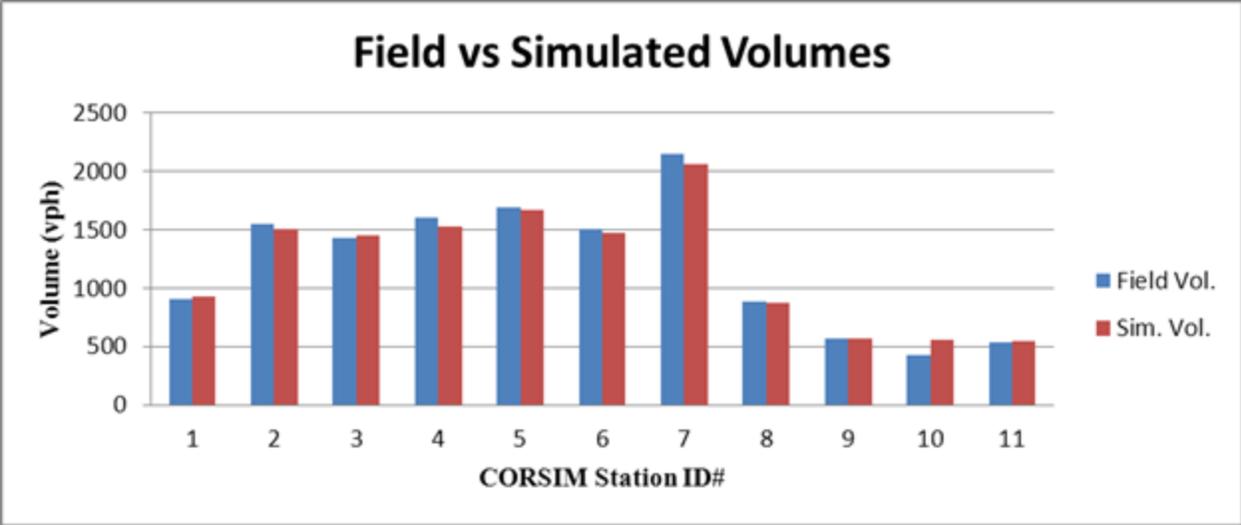


Figure A-11. Volume calibration for time period 11

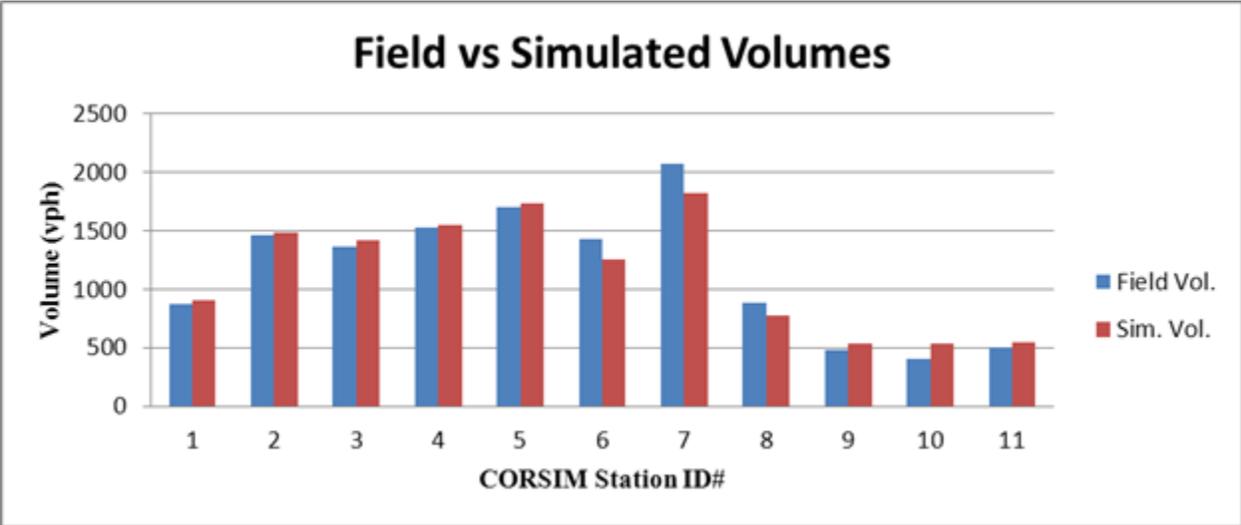


Figure A-12. Volume calibration for time period 12

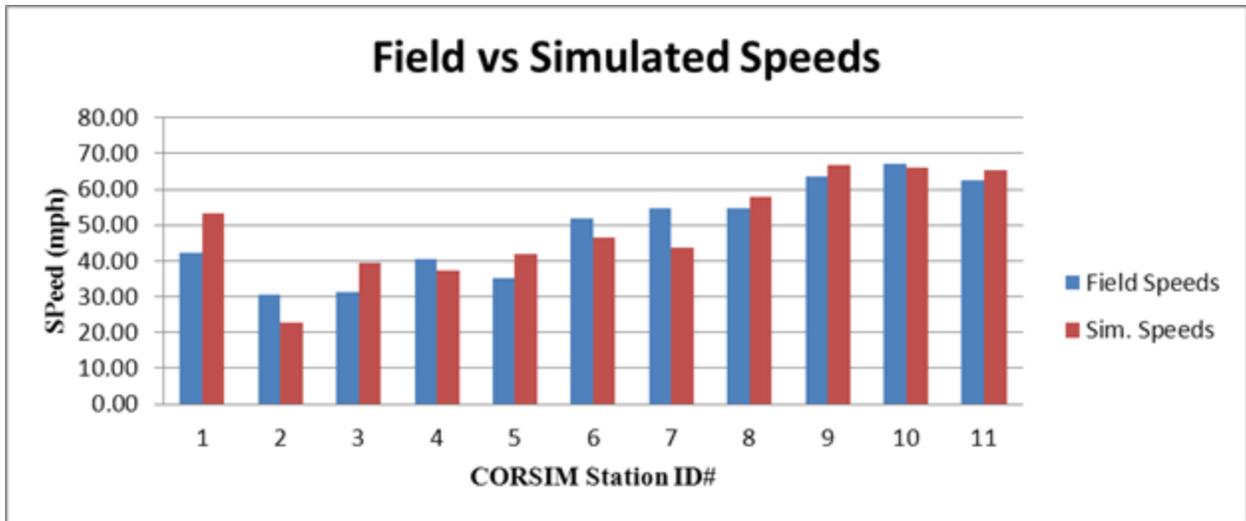


Figure A-13. Speed calibration for time period 1

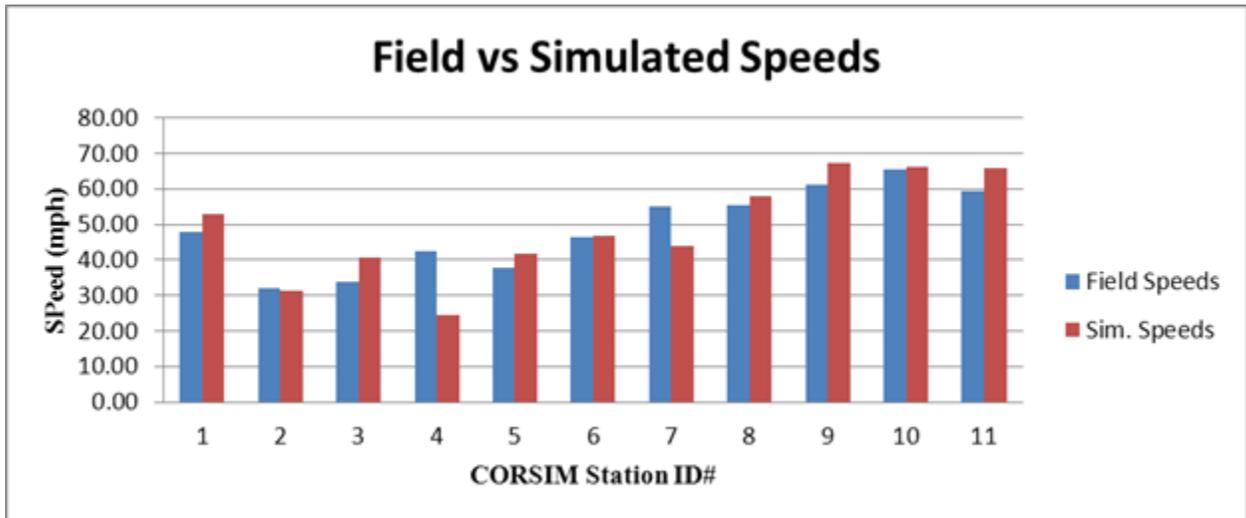


Figure A-14. Speed calibration for time period 2

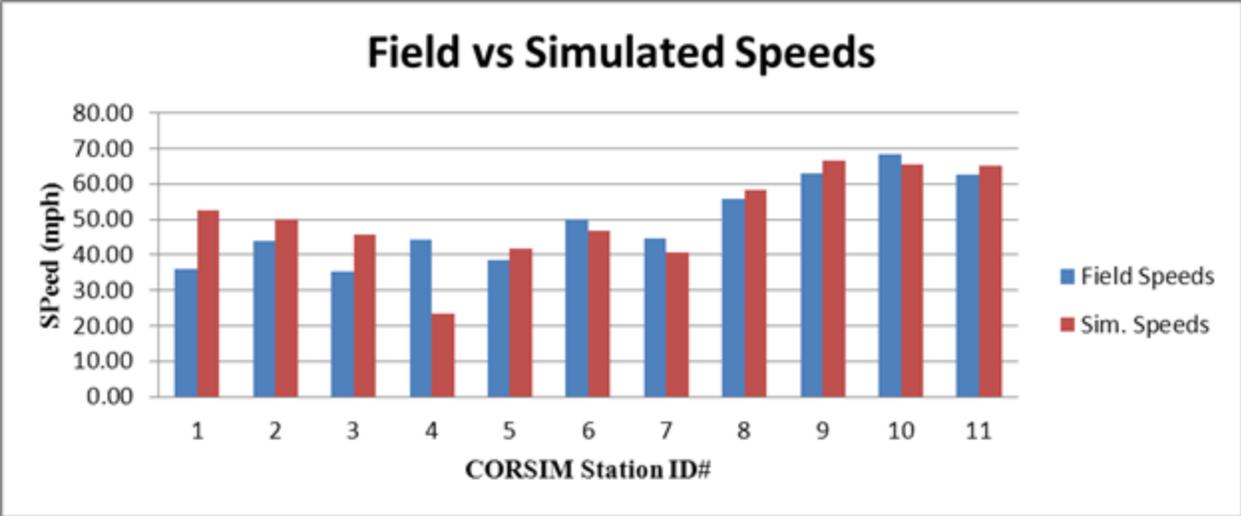


Figure A-15. Speed calibration for time period 3

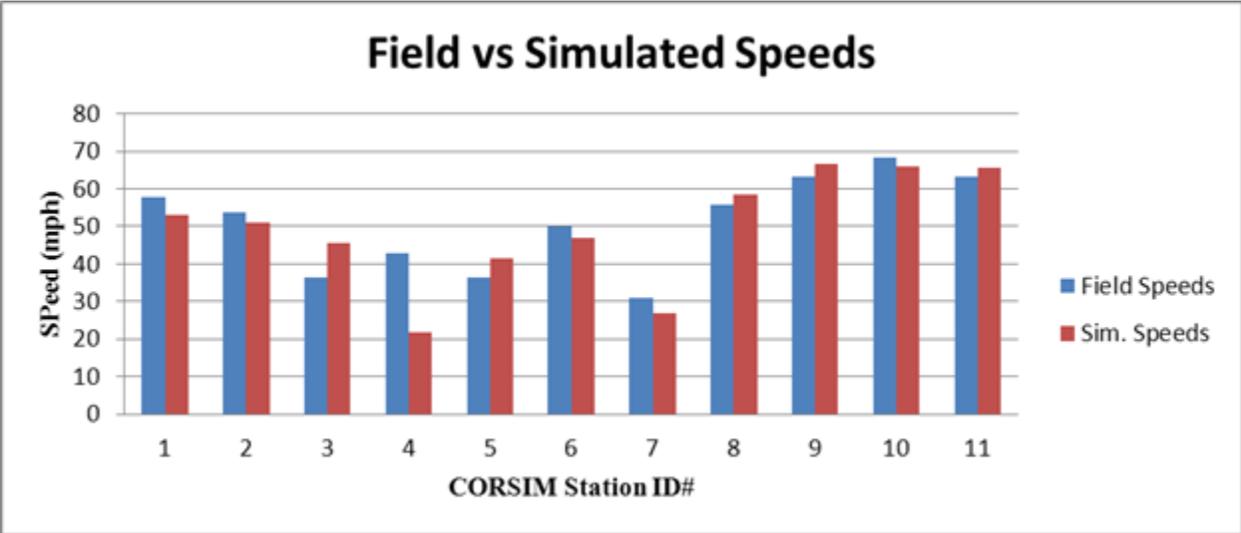


Figure A-16. Speed calibration for time period 4

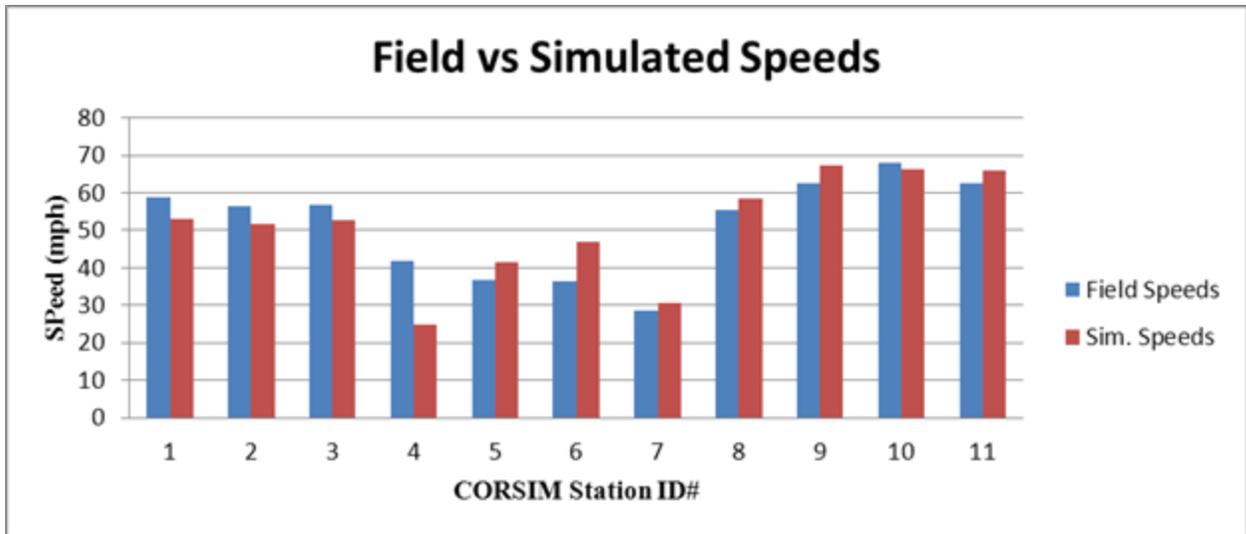


Figure A-17. Speed calibration for time period 5

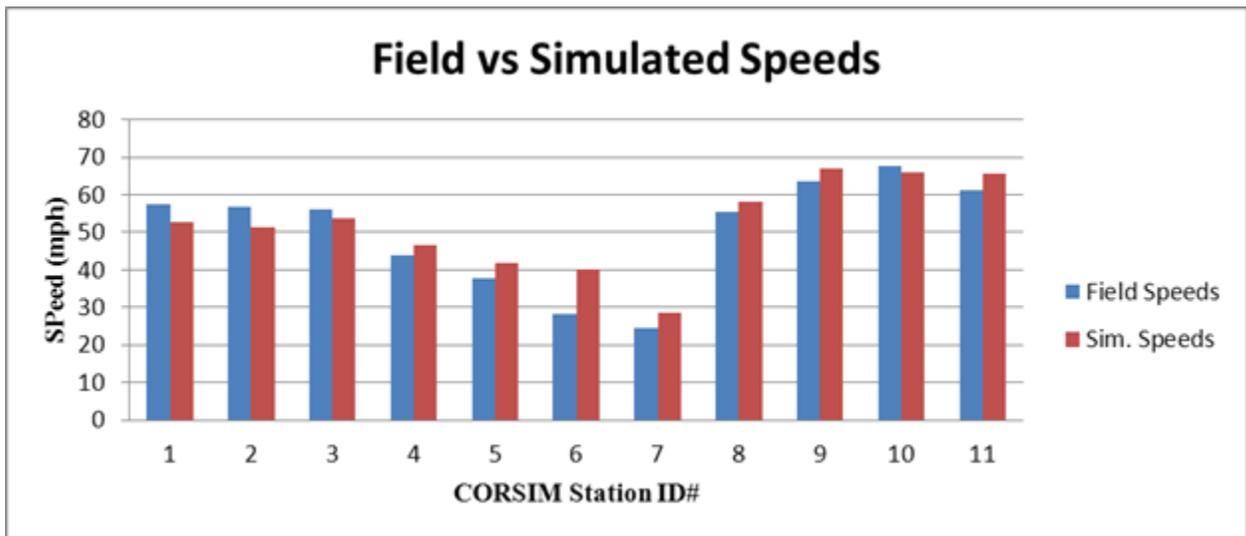


Figure A-18. Speed calibration for time period 6

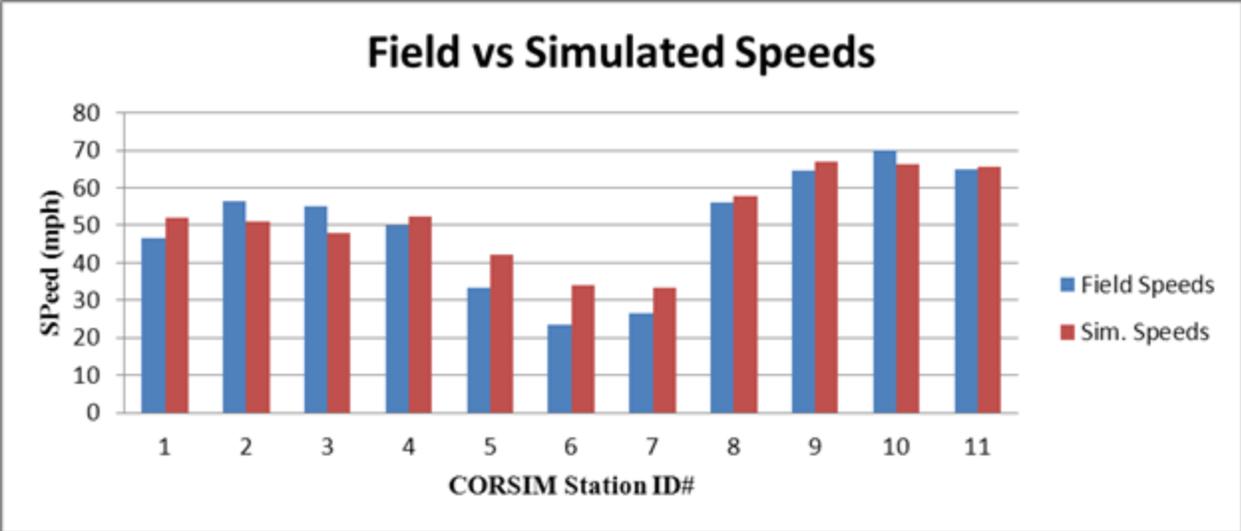


Figure A-19. Speed calibration for time period 7

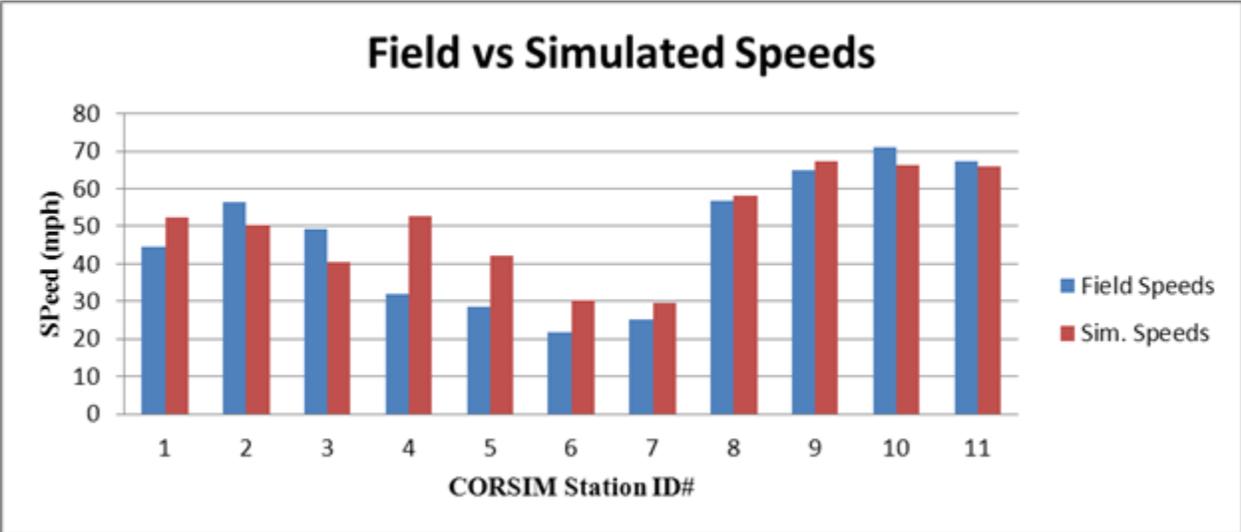


Figure A-20. Speed calibration for time period 8

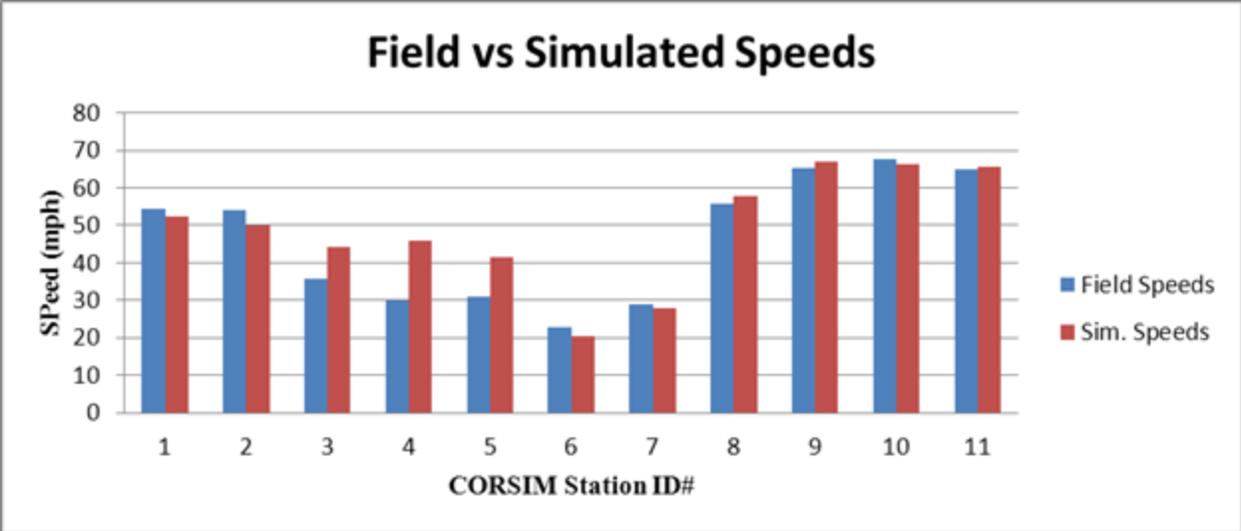


Figure A-21. Speed calibration for time period 9

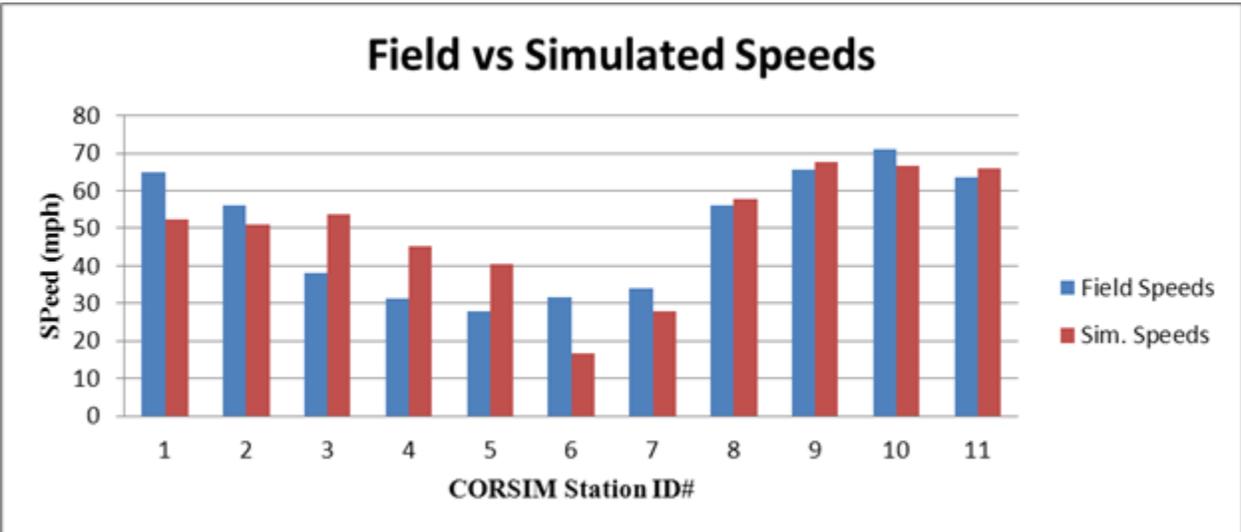


Figure A-22. Speed calibration for time period 10

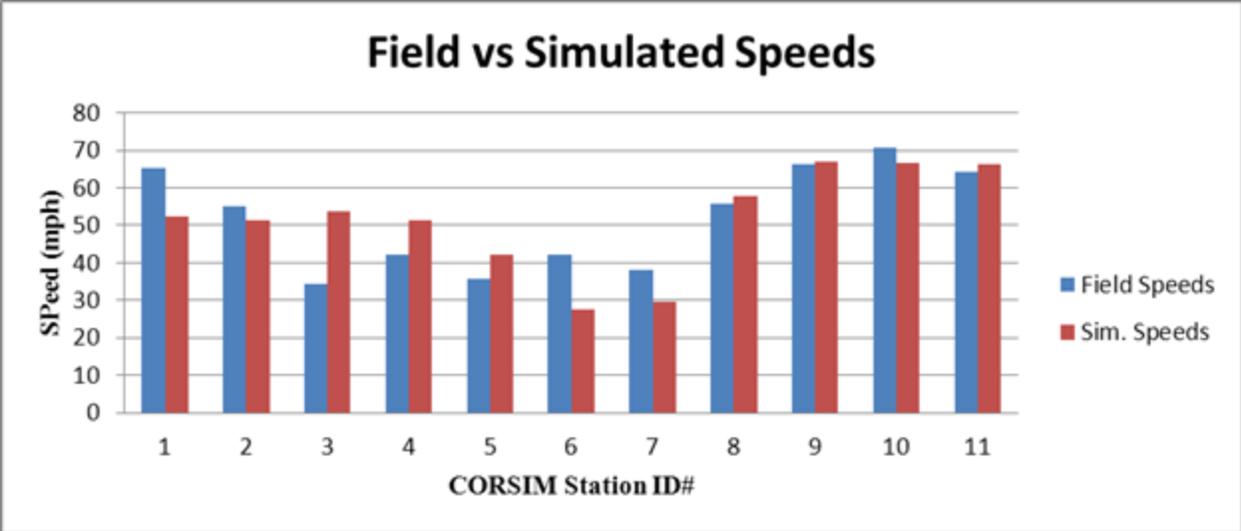


Figure A-23. Speed calibration for time period 11

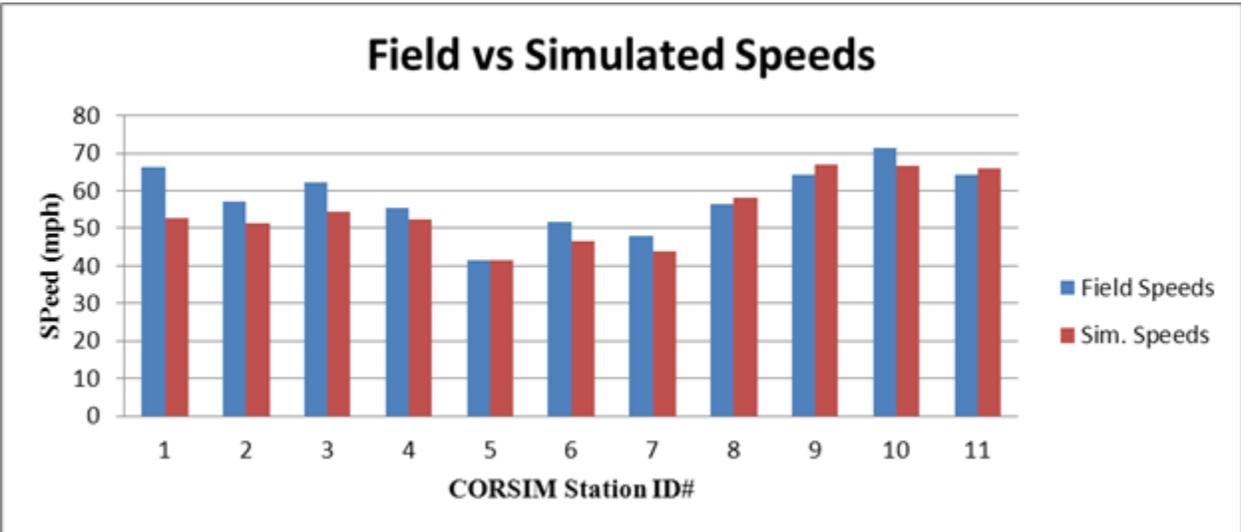


Figure A-24. Speed calibration for time period 12

APPENDIX B
SAMPLE SOURCE CODE FOR RTE

This appendix gives the source code for the RTE of one scenario. This is the occupancy-based algorithm using 2 signs spaced ½ mile apart using threshold scenario 1. The header file that imports and redefines variables is shown first, followed by the main DLL source code.

```
ifndef _FRESIM_H_
#define _FRESIM_H_

#define DLL_IMPORT extern "C" __declspec( dllimport )
#define DLL_EXPORT extern "C" __declspec( dllexport )

#define IMXNOD 8999

DLL_IMPORT struct { int YINIT; }          GLR091;
#define yinit                             GLR091.YINIT

DLL_IMPORT struct { float ZCLOCK; }      PRI306;
#define zclock                             P          RI306.ZCLOCK

DLL_IMPORT struct { int TTLFLK; }       PRI215;
#define ttlflk                             PRI215.TTLFLK

DLL_IMPORT struct { int NFMAP[IMXNOD]; } PRI075;
#define nfmap                             PRI075.NFMAP

DLL_IMPORT struct { int DPPINT; }       GDET01;
#define dppint                             GDET01.DPPINT

DLL_IMPORT float* FRESIM_DETECTORS_mp_ZFDOCC;
#define zfdocc FRESIM_DETECTORS_mp_ZFDOCC

DLL_IMPORT float* FRESIM_DETECTORS_mp_ZFDSPD;
#define zfdspd FRESIM_DETECTORS_mp_ZFDSPD

DLL_IMPORT float* FRESIM_DETECTORS_mp_ZFDVOL;
#define zfdvol FRESIM_DETECTORS_mp_ZFDVOL

DLL_IMPORT float* FRESIM_LINKS_mp_ZFFLOW;
#define zfflow FRESIM_LINKS_mp_ZFFLOW

DLL_IMPORT int* FRESIM_LINKS_mp_DWNODC;
#define dwnodc FRESIM_LINKS_mp_DWNODC
```

```
DLL_IMPORT int* FRESIM_LINKS_mp_UPNODC;  
#define upnodc FRESIM_LINKS_mp_UPNODC  
  
DLL_IMPORT int* FRESIM_LINKS_mp_DWNODE;  
#define dwnode FRESIM_LINKS_mp_DWNODE  
  
DLL_IMPORT int* FRESIM_LINKS_mp_UPNODE;  
#define upnode FRESIM_LINKS_mp_UPNODE  
  
DLL_IMPORT int* FRESIM_DETECTORS_mp_DETLK;  
#define detlk FRESIM_DETECTORS_mp_DETLK  
  
DLL_IMPORT int* GLOBAL_LINKS_mp_INMAP;  
#define inmap GLOBAL_LINKS_mp_INMAP  
  
DLL_IMPORT int* FRESIM_LINKS_mp_FDETID;  
#define fdetid FRESIM_LINKS_mp_FDETID  
  
DLL_IMPORT int* FRESIM_DETECTORS_mp_NDETLK;  
#define ndetlk FRESIM_DETECTORS_mp_NDETLK  
  
#endif //_FRESIM_H_
```

```

// This is the main DLL file.

#include "stdafx.h"
#include "VSL.h"
#include "CORWin.h"
#include "fresim.h"

// Declare Functions and Global Variables

int getDet(int link, int config);
int getLinks(int up, int down);
void updateSpeed(int link, int d1, int d2, int d3, int d4, int d5, int d6, int d7, int d8, int d9);
void moveSpeeds(int lnk, float spd[], int maxTime);
void upSign(int lnk, float spd[], int maxTime);

int L159L161, L161L165, L165L166, L166L167, L158L159, L258L158, L154L155,
L155L156, L156L157, L157L158;
int d1, d2, d3, d4, d5;
float evalTime = 60.000000, horizonTime = 60.000000;
float speed[10] = {73, 73, 73, 73, 73, 73, 73, 73, 73, 73};
float upSpeed[10] = {73, 73, 73, 73, 73, 73, 73, 73, 73, 73};

DLL_EXPORT void _stdcall vsl_Initialize(){

    // Reassigning link numbers to match CORSIM Internal Link Numbering to
    // User Defined Link Numbering getLinks(upstream node, downstream node)

        L154L155 = getLinks(154, 155);
        L155L156 = getLinks(155, 156);
        L156L157 = getLinks(156, 157);
        L157L158 = getLinks(157, 158);
        L158L159 = getLinks(158, 159);
        L159L161 = getLinks(159, 161);
        L161L165 = getLinks(161, 165);

    // Assigning detector numbering getDet(link, lane)

        d1 = getDet(L159L161, 1);
        d2 = getDet(L159L161, 2);
        d3 = getDet(L159L161, 3);
        d4 = getDet(L159L161, 4);
        d5 = getDet(L159L161, 5);

    // Setting the Point processing Interval
    dppint = 60;

```

```

}

DLL_EXPORT void _stdcall vsl_PrefreesimVehicle()
{
    if (yinit != 1){

        if (zclock == evalTime){
            evalTime = evalTime + 60;
            updateSpeed(L158L159, d1, d2, d3, d4, d5, 9999, 0, 0, 0);
        }

        if (zclock == horizonTime){
            horizonTime = horizonTime + 15;

            moveSpeeds(L158L159, speed, 9);
            zfflow[L159L161] = speed[1] + 7; zfflow[L161L165] = speed[2] + 7;

            upSign(L154L155, upSpeed, 9);
            zfflow[L154L155] = upSpeed[0]; zfflow[L155L156] = upSpeed[1];
            zfflow[L156L157] = upSpeed[2]; zfflow[L157L158] = upSpeed[4];
        }
    }
}

int getLinks(int up, int down){
    int dnode = 0;
    int unode = 0;
    int LinkID = 0;

    for (int index = 0; index < ttlflk; index++){
        int dnode = dwnode[index];
        int unode = upnode[index];

        if (dnode < 7000) {dnode = nfmap[dnode-1];}
        if (unode < 7000) {unode = nfmap[unode-1];}

        if (up == unode && down == dnode){
            LinkID = index;
        }
    }
    return LinkID;
}

int getDet(int link, int config){
    int detectorID = 0;

```

```

if (fdetid[link] != 0){
    if (config == 1){
        detectorID = fdetid[link] - 1;
    }
    if (config == 2){
        detectorID = (fdetid[link] - 1) + 2;
    }
    if (config == 3){
        detectorID = (fdetid[link] - 1) + 4;
    }
    if (config == 4){
        detectorID = (fdetid[link] - 1) + 6;
    }
    if (config == 5){
        detectorID = (fdetid[link] - 1) + 8;
    }
    if (config == 6){
        detectorID = (fdetid[link] - 1) + 10;
    }
    if (config == 7){
        detectorID = (fdetid[link] - 1) + 12;
    }
    if (config == 8){
        detectorID = (fdetid[link] - 1) + 14;
    }
    if (config == 9){
        detectorID = (fdetid[link] - 1) + 16;
    }
}
return detectorID;
}

```

```

void updateSpeed(int link, int d1, int d2, int d3, int d4, int d5, int d6, int d7, int d8, int d9){
    int n = 0;
    int detNum = 0;
    float average = 0;
    if (d9 == 9999){ detNum = 8;}
    if (d8 == 9999){ detNum = 7;}
    if (d7 == 9999){ detNum = 6;}
    if (d6 == 9999){ detNum = 5;}
    if (d5 == 9999){ detNum = 4;}
    if (d4 == 9999){ detNum = 3;}
    if (d3 == 9999){ detNum = 2;}
    if (d2 == 9999){ detNum = 1;}
}

```

```

switch (detNum) {
    case 0: average = (zfdocc[d1] + zfdocc[d2] + zfdocc[d3] + zfdocc[d4] +
        zfdocc[d5] + zfdocc[d6] + zfdocc[d7] + zfdocc[d8] +
        zfdocc[d9])/9;
        break;
    case 8: average = (zfdocc[d1] + zfdocc[d2] + zfdocc[d3] + zfdocc[d4] +
        zfdocc[d5] + zfdocc[d6] + zfdocc[d7] + zfdocc[d8])/8;
        break;
    case 7: average = (zfdocc[d1] + zfdocc[d2] + zfdocc[d3] + zfdocc[d4] +
        zfdocc[d5] + zfdocc[d6] + zfdocc[d7])/7;
        break;
    case 6: average = (zfdocc[d1] + zfdocc[d2] + zfdocc[d3] + zfdocc[d4] +
        zfdocc[d5] + zfdocc[d6])/6;
        break;
    case 5: average = (zfdocc[d1] + zfdocc[d2] + zfdocc[d3] + zfdocc[d4] +
        zfdocc[d5])/5;
        break;
    case 4: average = (zfdocc[d1] + zfdocc[d2] + zfdocc[d3] + zfdocc[d4])/4;
        break;
    case 3: average = (zfdocc[d1] + zfdocc[d2] + zfdocc[d3])/3;
        break;
    case 2: average = (zfdocc[d1] + zfdocc[d2])/2;
        break;
    case 1: average = zfdocc[d1];
        break;
}

If (zfflow[link] >73) n = 0;
else if (zfflow[link] >62 && zfflow[link] < 70) n = 1;
else if (zfflow[link] < 60) n = 2;
switch (n) {
    case 0:
        if(average > 16){
            zfflow[link] = 66;
            char text[132];
            sprintf_s( text, "Speed Limit has been reduced to 50 MPH, with
                Average Occupancy:%f\n", average);
            OutputString( text, 132, 2, 0 );
        }
        else {
            zfflow[link] = 73;
        }
        break;
}

```

```

case 1:
    if(average <= 12){
        zfflow[link] = 73;
        char text[132];
        sprintf_s( text, "Speed Limit has been Increased back to 55 MPH,
                        with Average Occupancy:%f\n", average);
        OutputString( text, 132, 2, 0 );
    }

    if (average > 12){

        if(average > 28){
            zfflow[link] = 58;
            char text[132];
            sprintf_s( text, "Speed Limit has been Reduced to 45 MPH,
                            with Average Occupancy:%f\n", average);
            OutputString( text, 132, 2, 0 );
        }

        if (average <= 28) {
            zfflow[link] = 66;
        }
    }
    break;

case 2:
    if (average > 25){
        zfflow[link] = 58;
    }

    else{
        zfflow[link] = 66;
        char text[132];
        sprintf_s( text, "Speed Limit has been Increased back to 50 MPH,
                        with Average Occupancy:%f\n", average);
        OutputString( text, 132, 2, 0 );
    }
    break;
}

}

void moveSpeeds(int lnk, float spd[], int maxTime){

    for (int i = maxTime; i > 0; i--){
        spd[i] = spd[i-1];
    }
}

```

```
        spd[0] = zfflow[lnk];
    }
void upSign(int lnk, float spd[], int maxTime){
    for (int i = maxTime; i > 0; i--){
        spd[i] = spd[i-1];
    }
    if (speed[0] < 60){
        spd[0] = speed[0] + 7;
    }
    else {
        spd[0] = 73;
    }
}
```

APPENDIX C SPEED PROFILES FOR SCENARIOS TESTED

This appendix displays the speed profiles over all 12 time periods for each scenario tested.

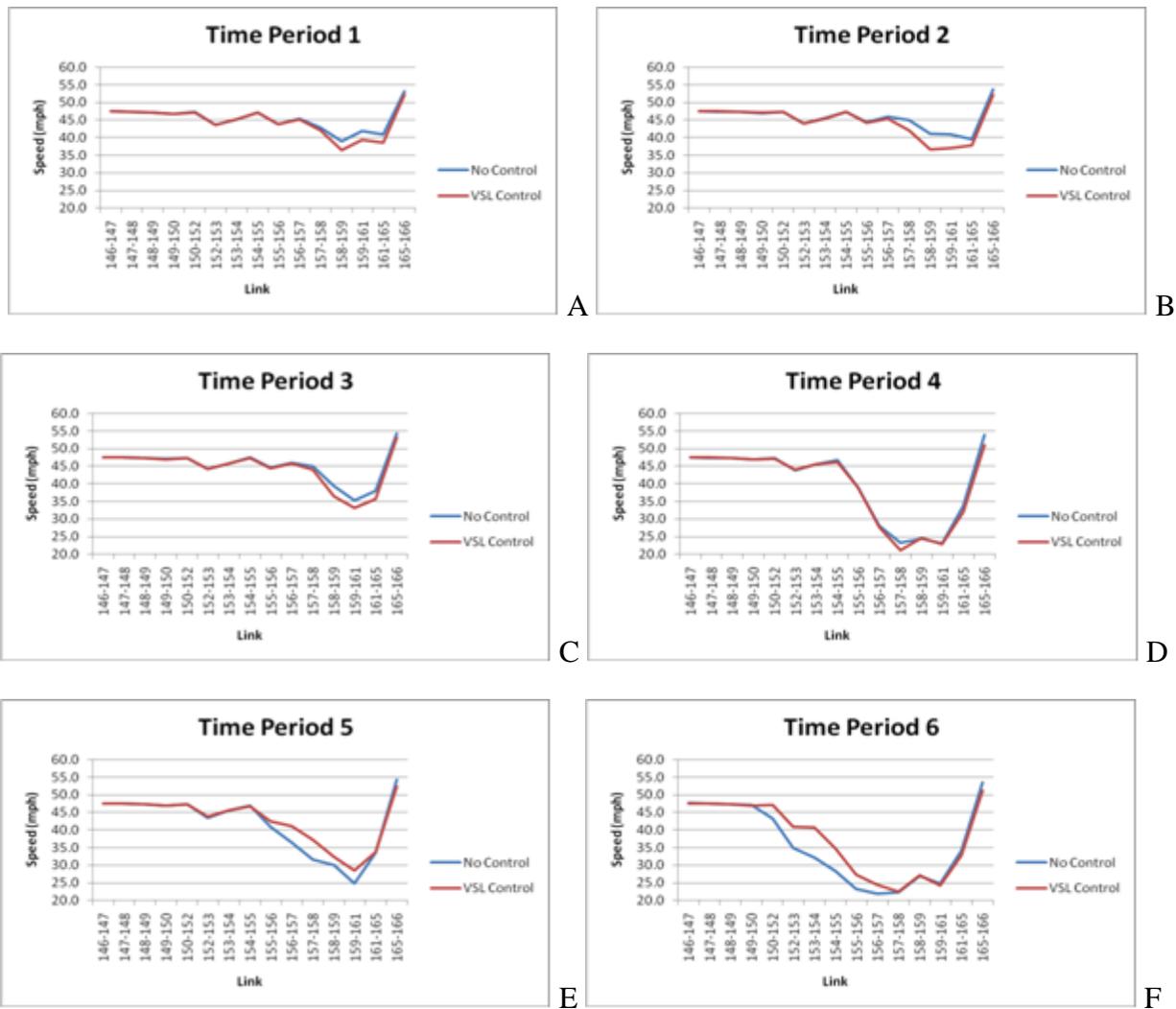
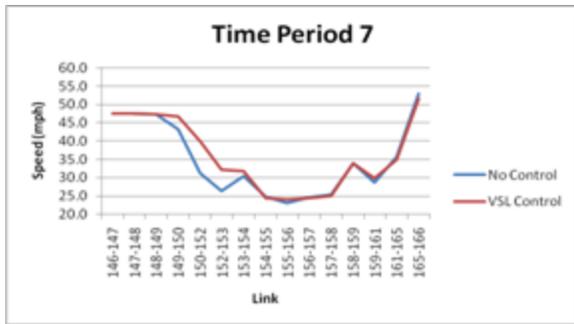
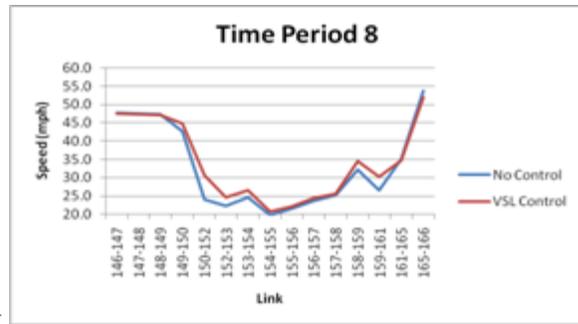


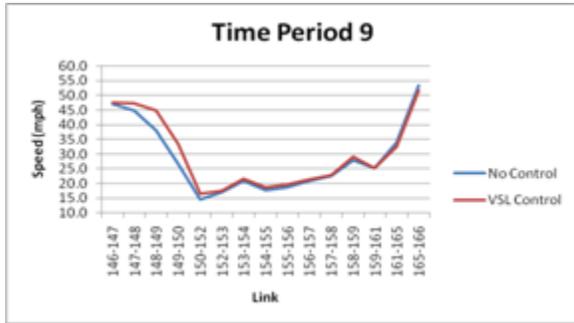
Figure C-1. Speed profile for occupancy-based algorithm using 1 sign spaced $\frac{1}{2}$ mile from the bottleneck source (threshold scenario 1) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time Period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



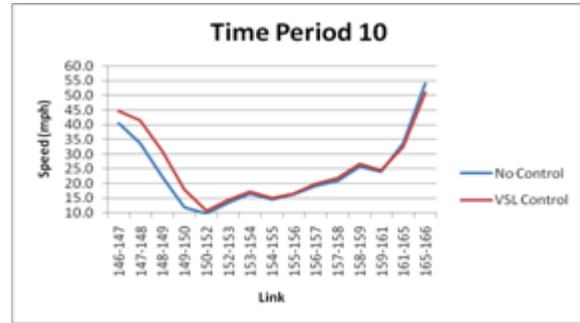
G



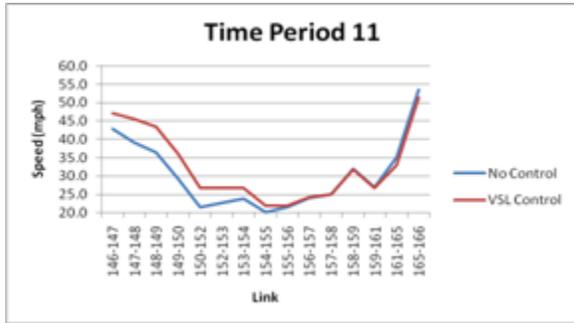
H



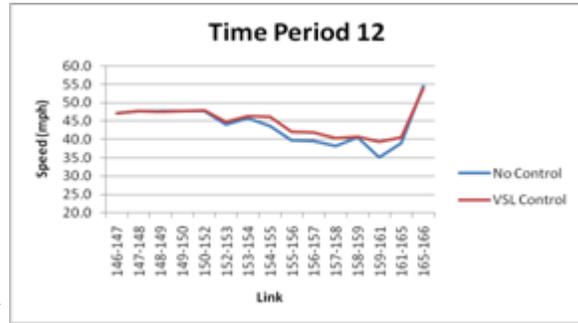
I



J



K



L

Figure C-1. Continued

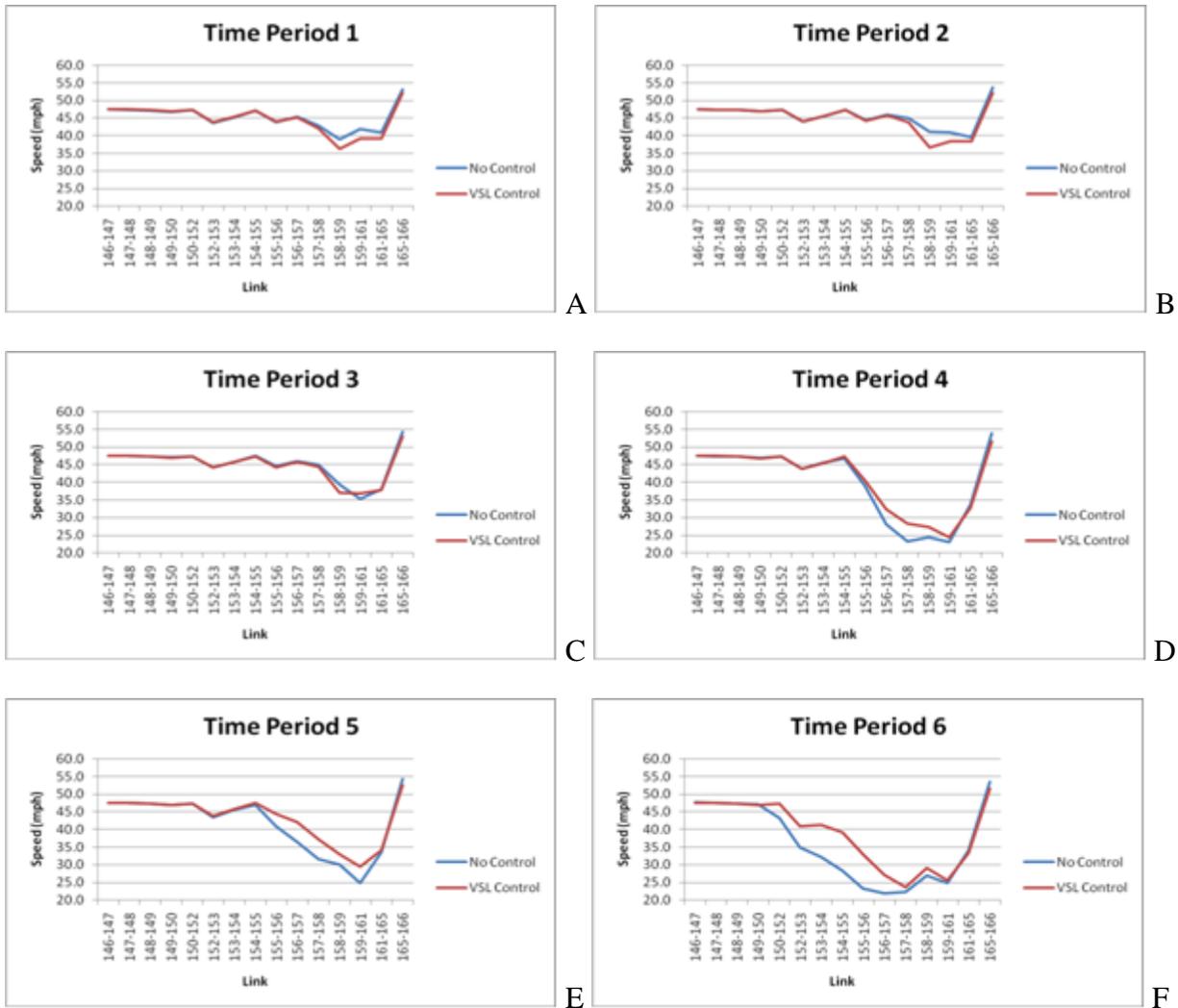
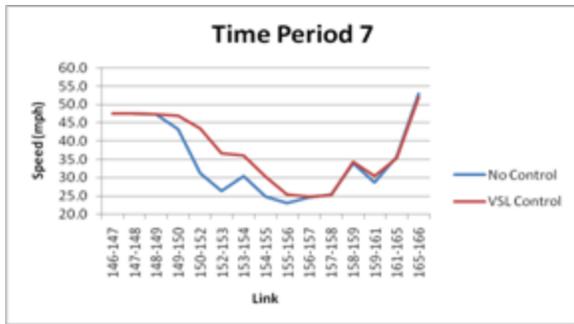
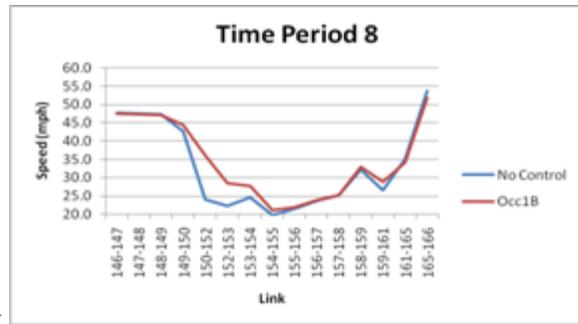


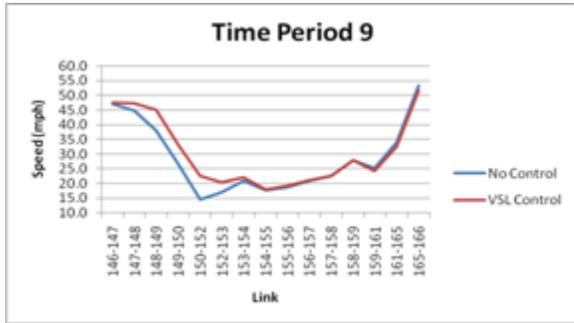
Figure C-2. Speed profile for occupancy-based algorithm using 1 sign spaced 1/2 mile from the bottleneck source (threshold scenario 2) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time Period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



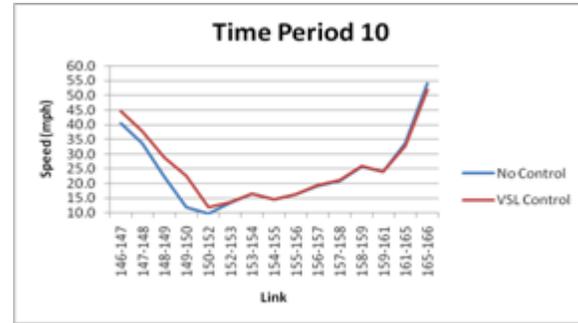
G



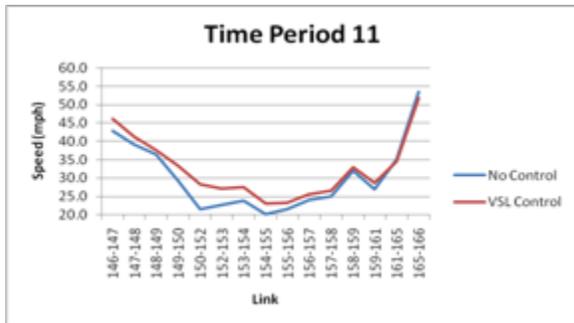
H



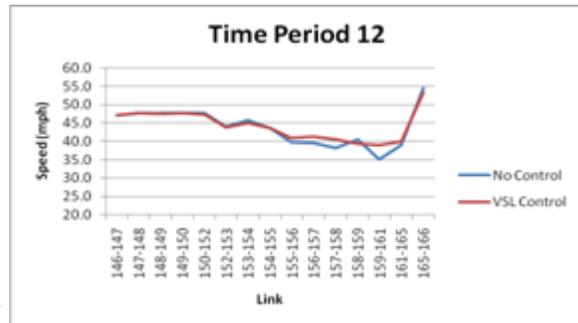
I



J



K



L

Figure C-2. Continued

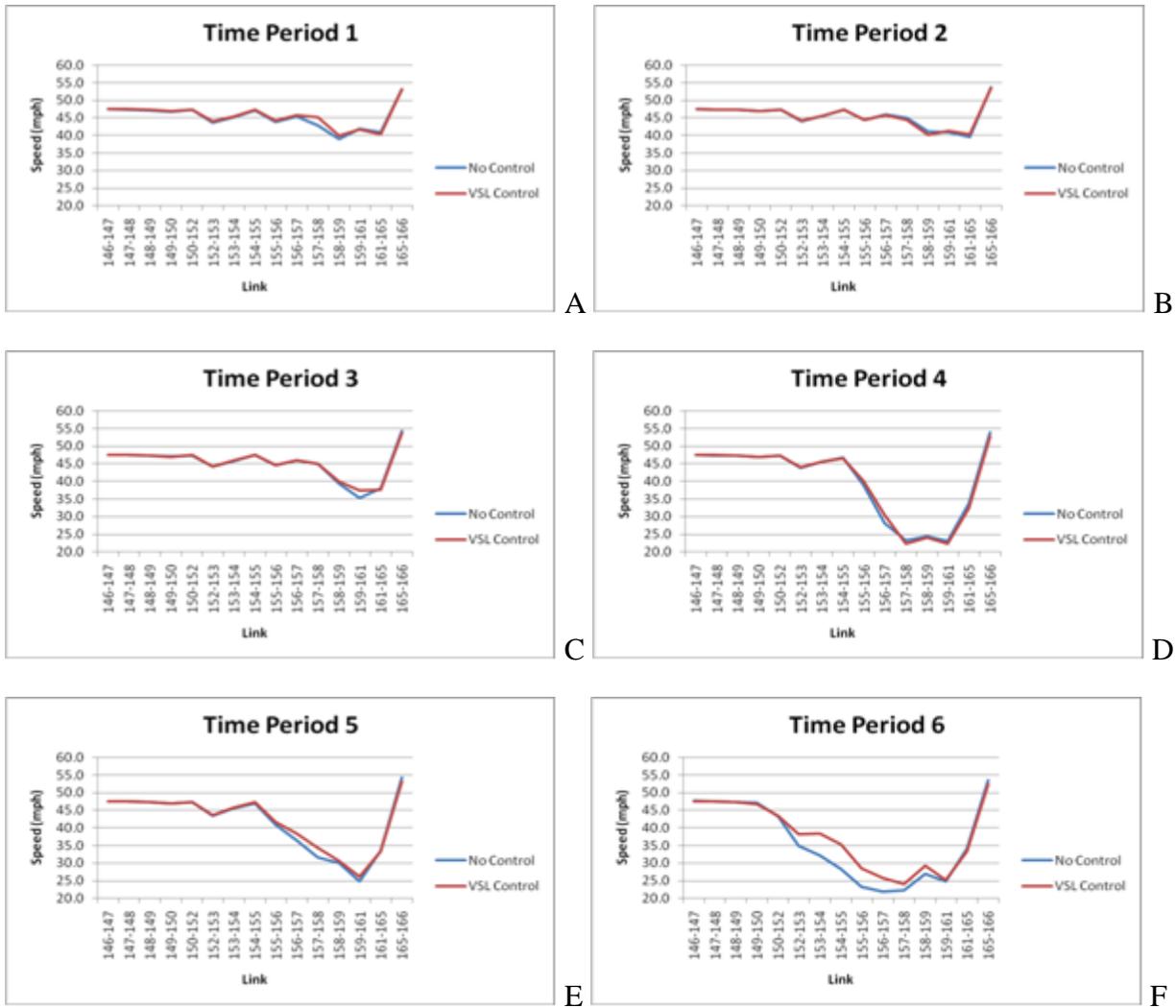
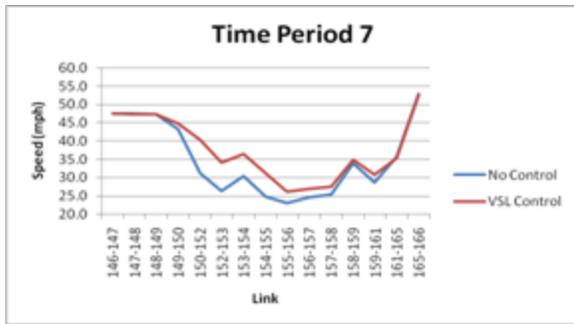
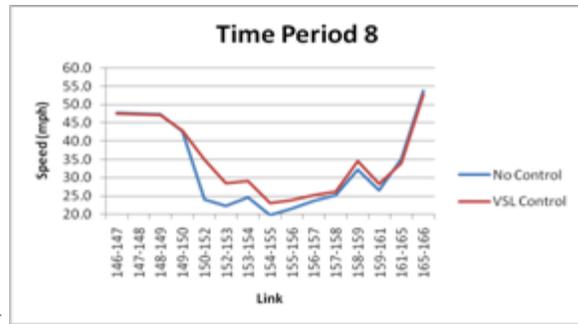


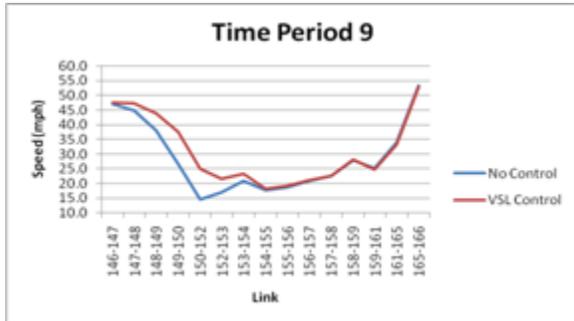
Figure C-3. Speed profile for occupancy-based algorithm using 1 sign spaced 1/2 mile from the bottleneck source (threshold scenario 3) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time Period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



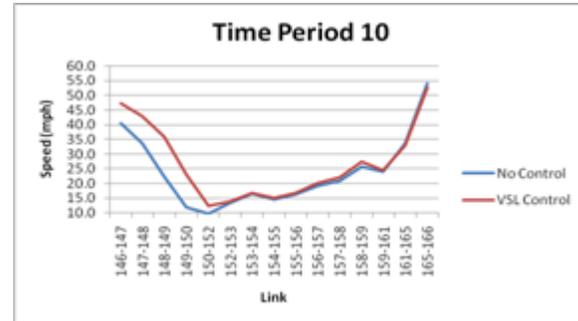
G



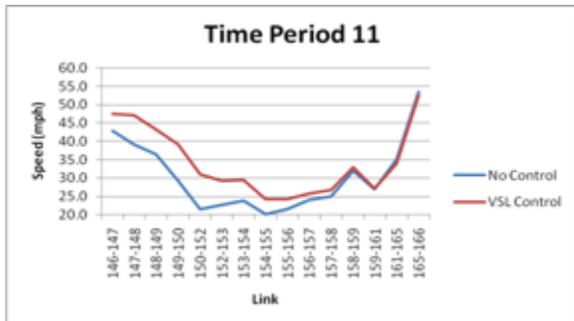
H



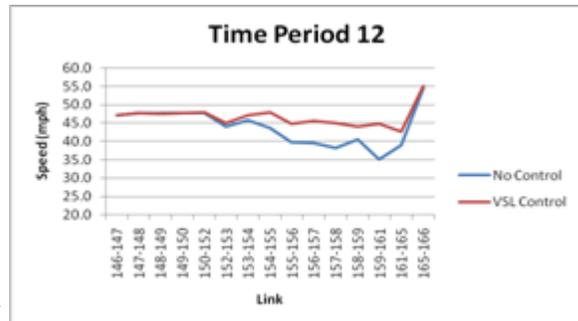
I



J



K



L

Figure C-3. Continued

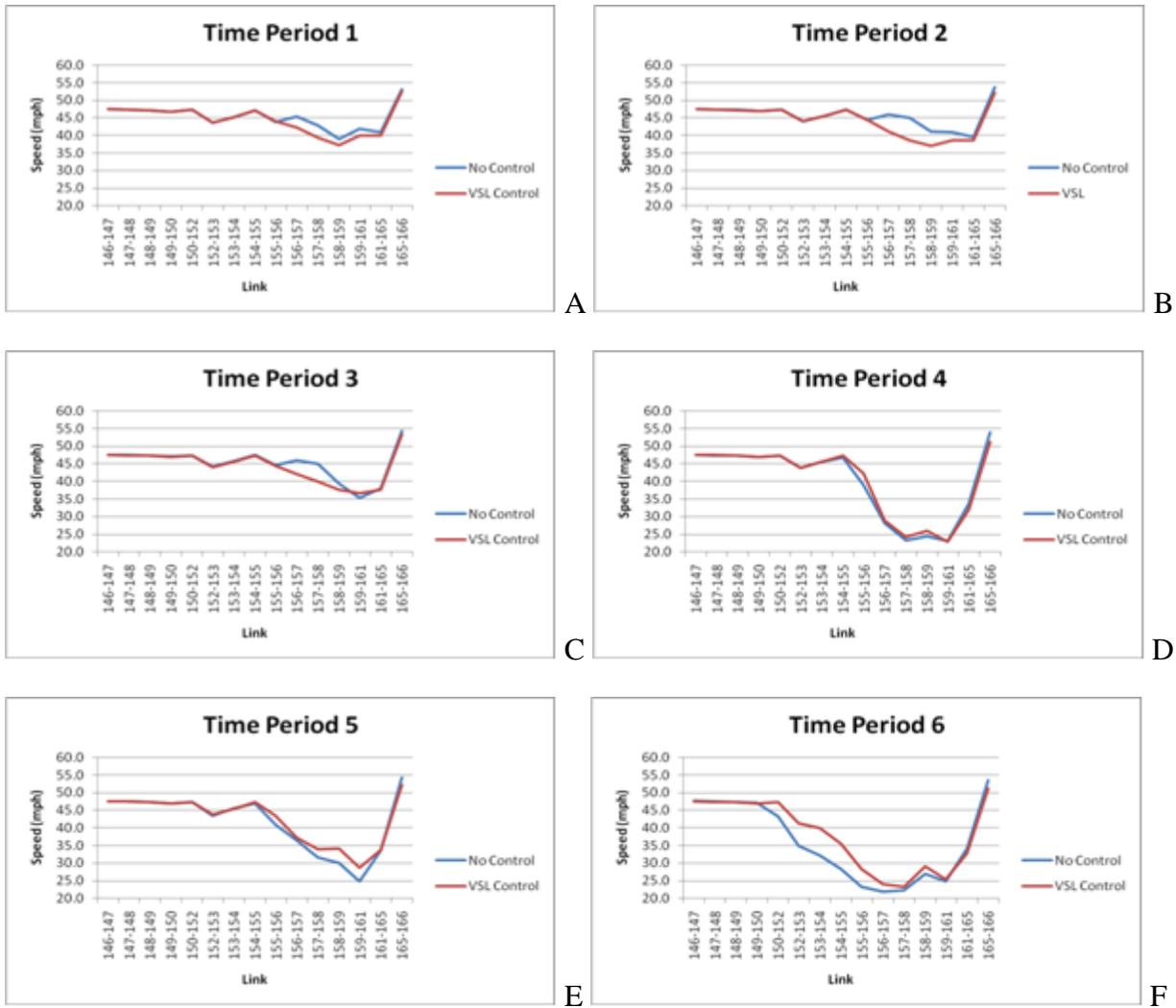
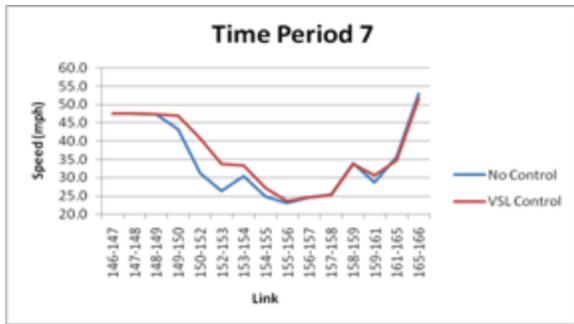
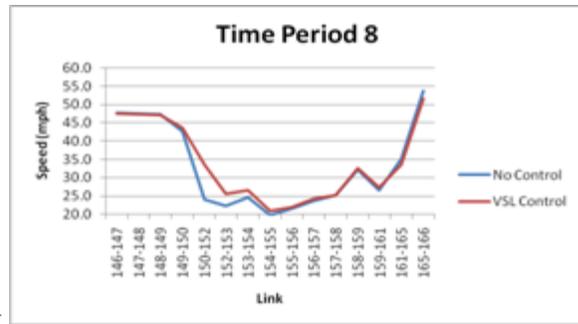


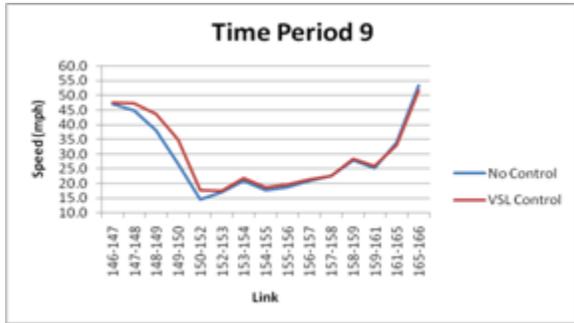
Figure C-4. Speed profile for occupancy-based algorithm using 1 sign spaced 1 mile from the bottleneck source (threshold scenario 1) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time Period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



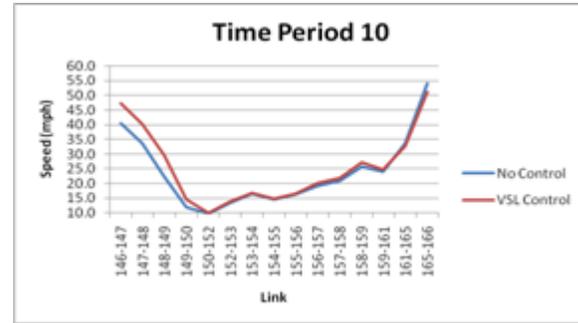
G



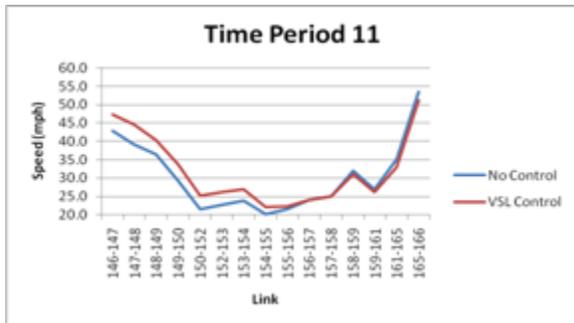
H



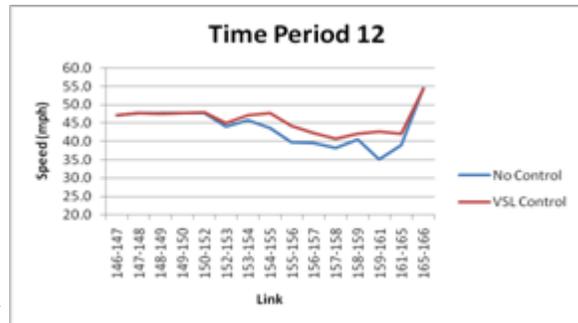
I



J



K



L

Figure C-4. Continued

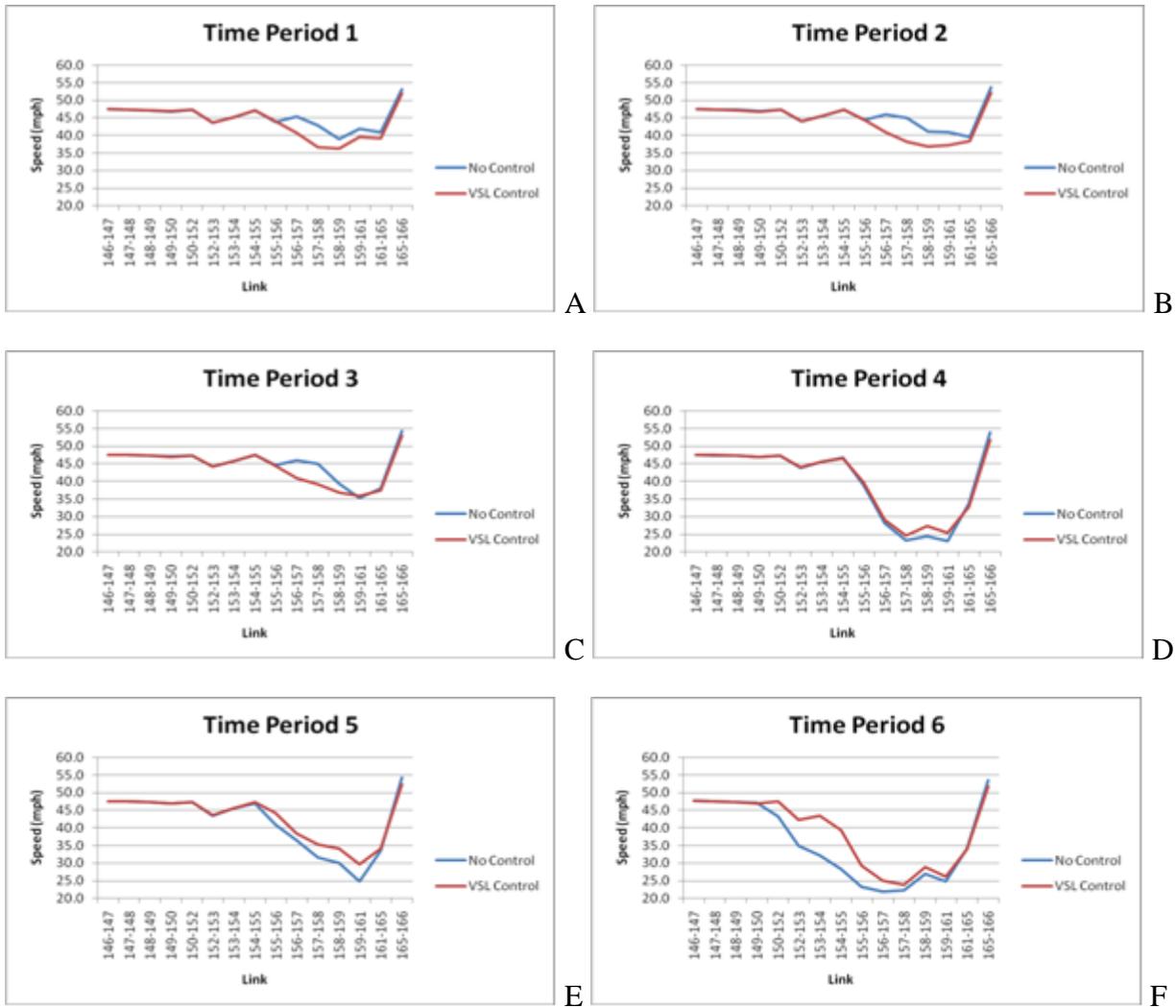
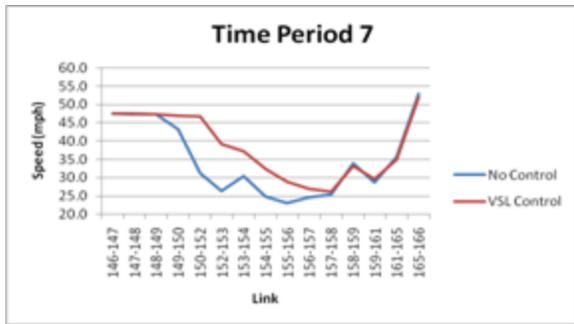
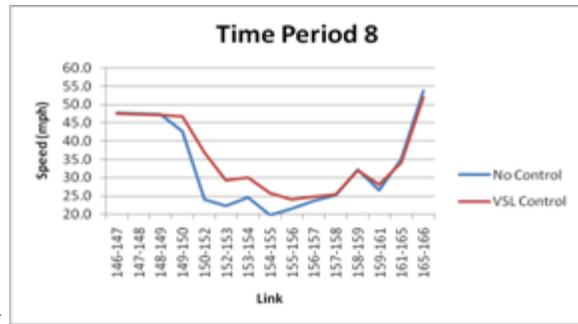


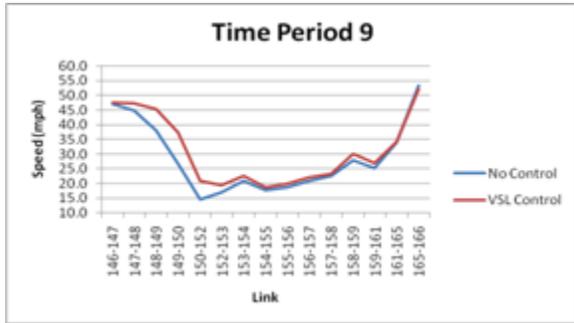
Figure C-5. Speed profile for occupancy-based algorithm using 1 sign spaced 1 mile from the bottleneck source (threshold scenario 2) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time Period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



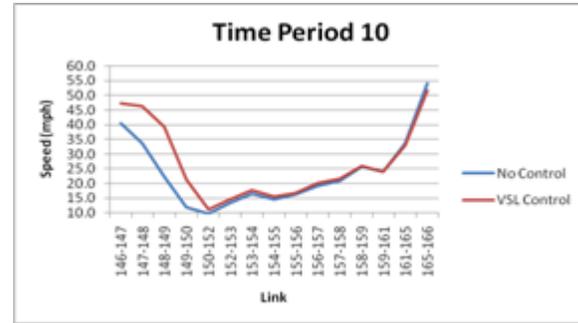
G



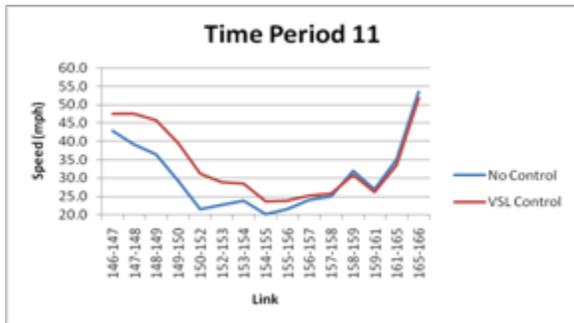
H



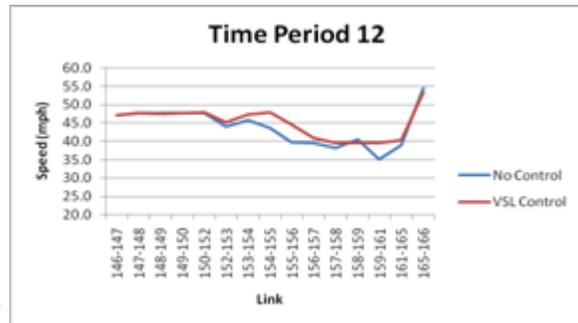
I



J



K



L

Figure C-5. Continued

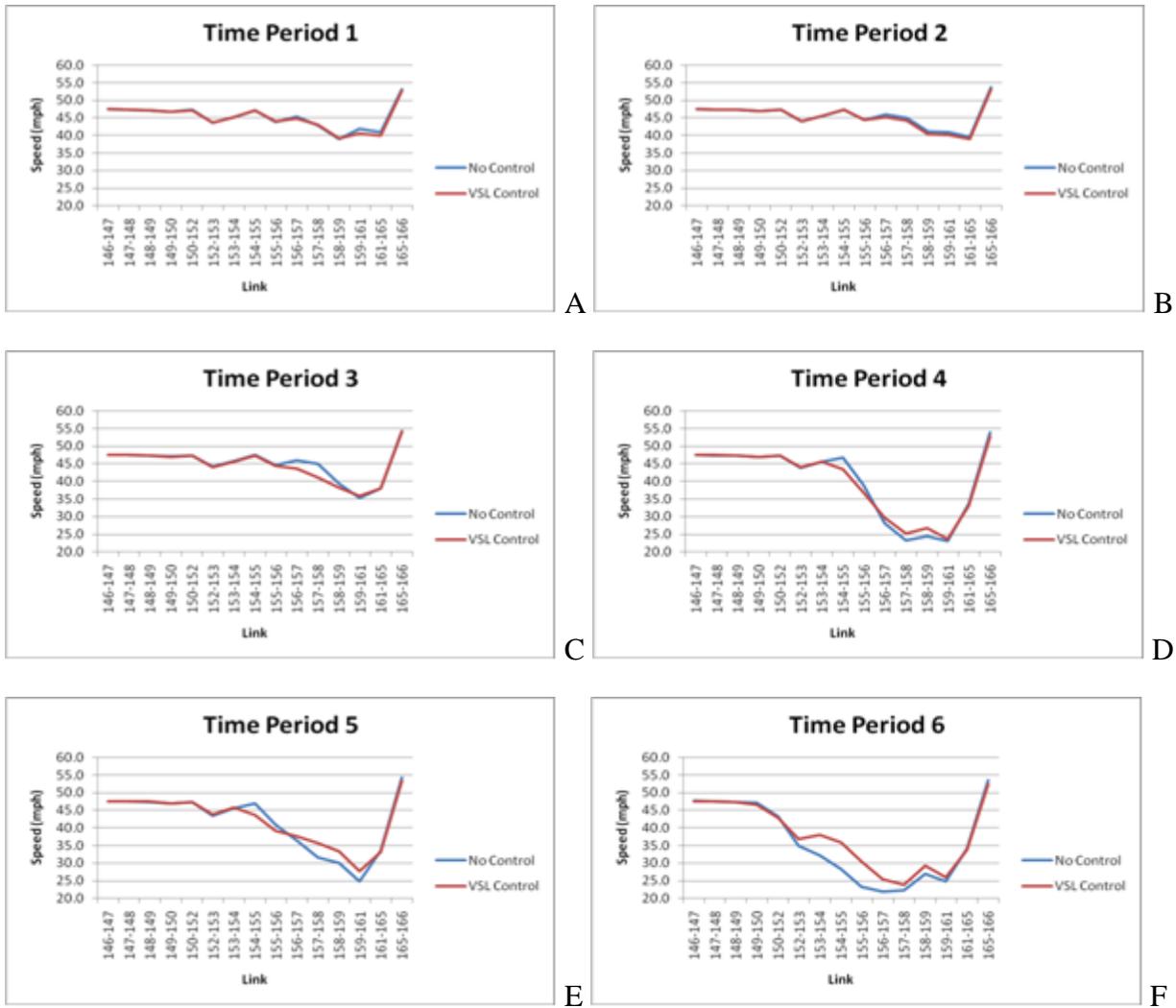
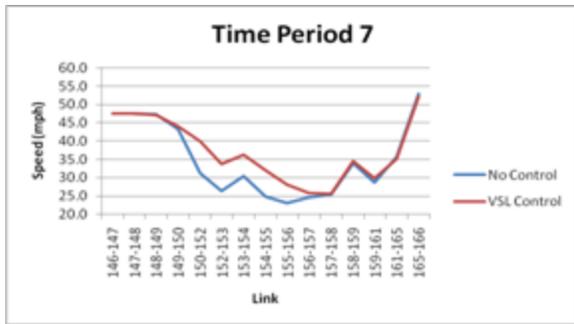
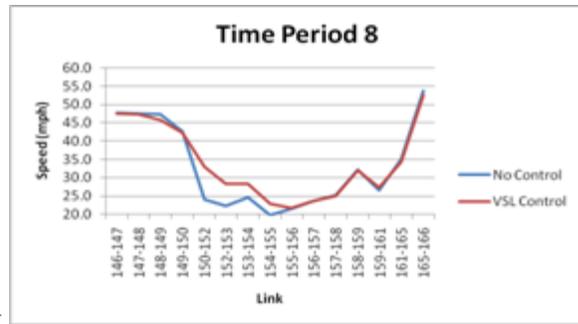


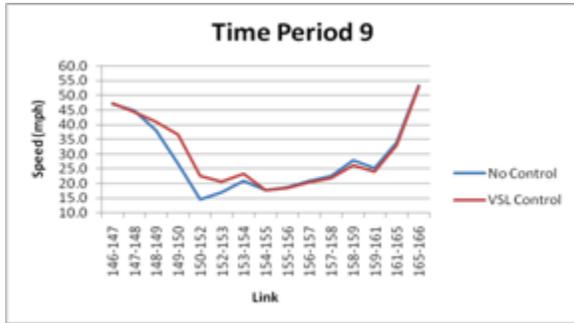
Figure C-6. Speed profile for occupancy-based algorithm using 1 sign spaced 1 mile from the bottleneck source (threshold scenario 3) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time Period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



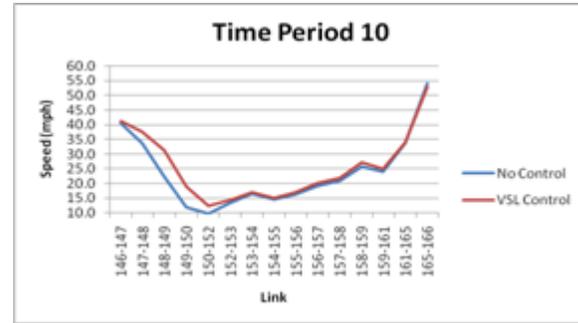
G



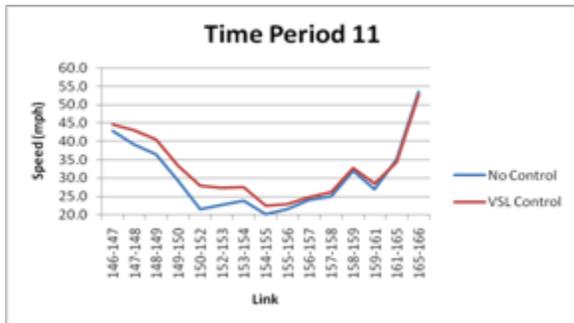
H



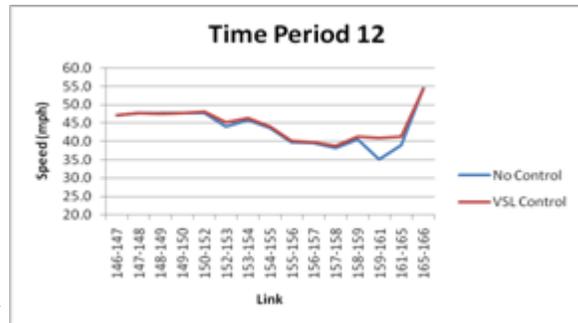
I



J



K



L

Figure C-6. Continued

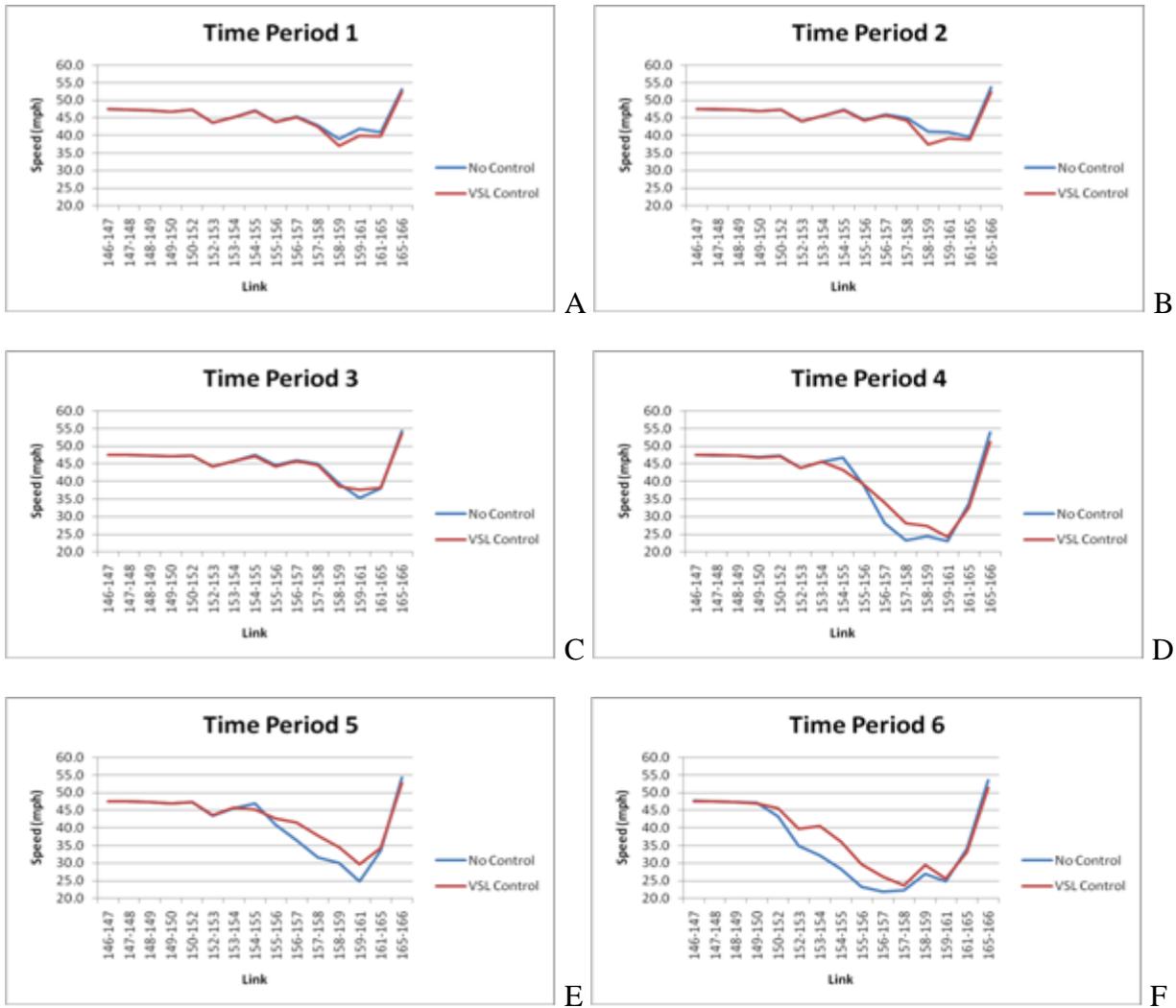
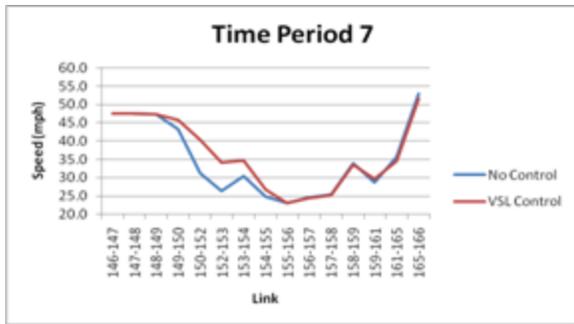
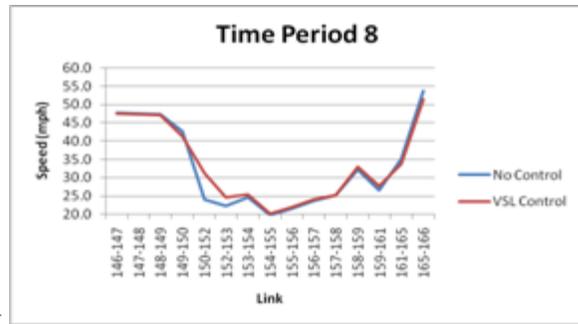


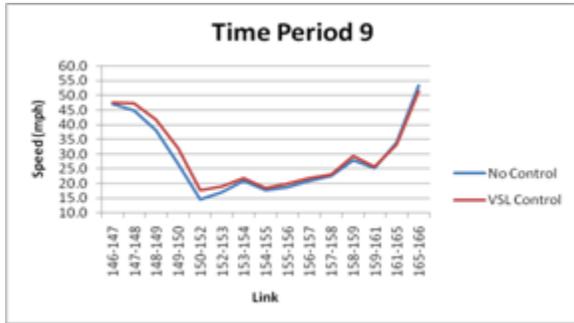
Figure C-7. Speed profile for occupancy-based algorithm using 2 signs spaced 1/2 mile apart (threshold scenario 1) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



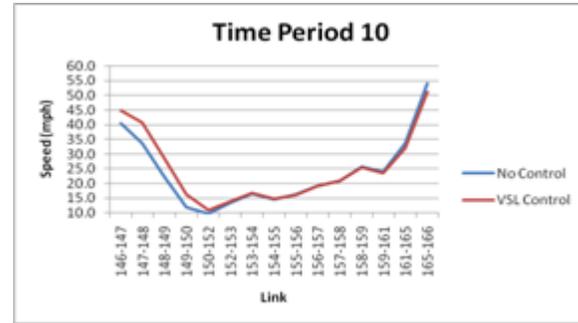
G



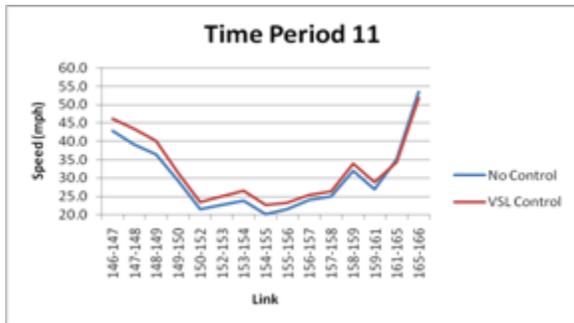
H



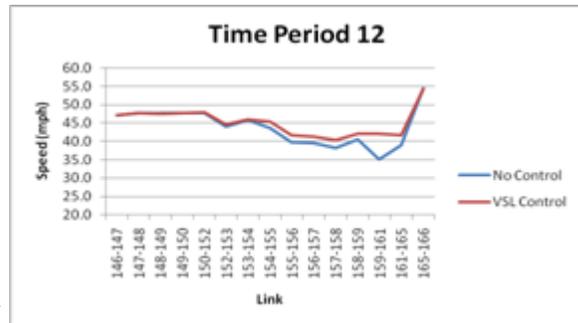
I



J



K



L

Figure C-7. Continued

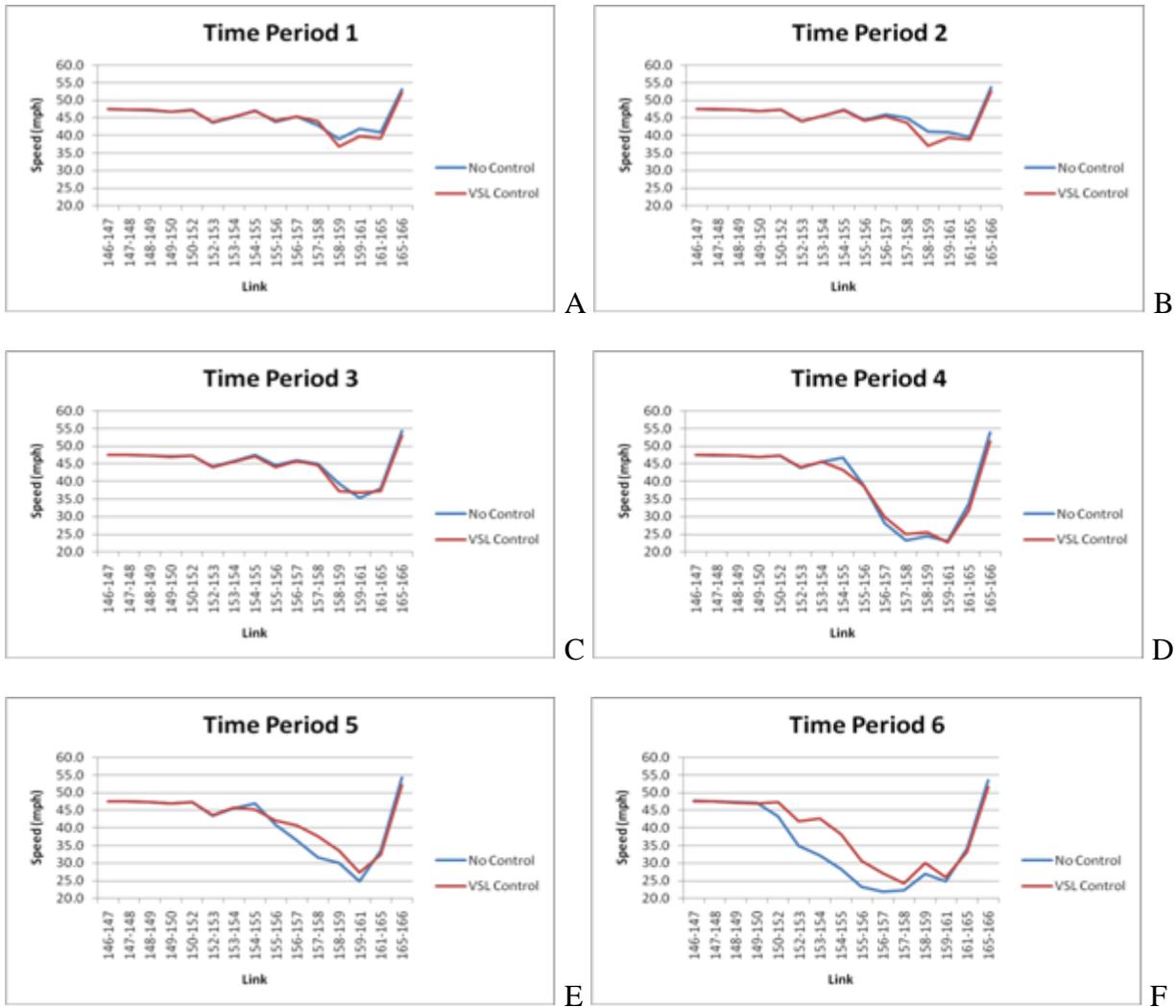
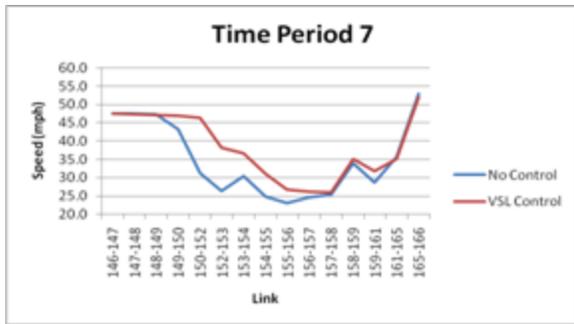
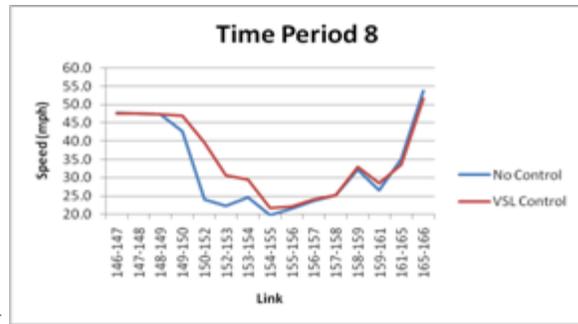


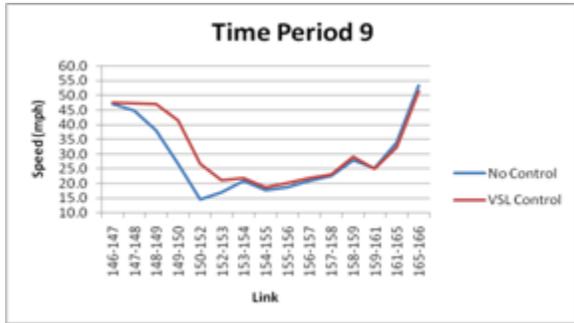
Figure C-8. Speed profile for occupancy-based algorithm using 2 signs spaced 1/2 mile apart (threshold scenario 2) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



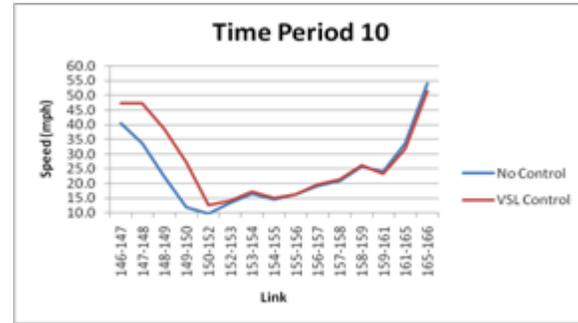
G



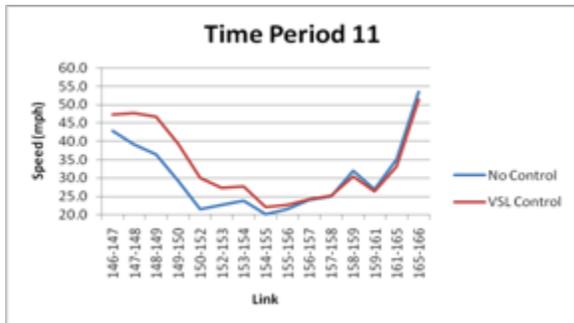
H



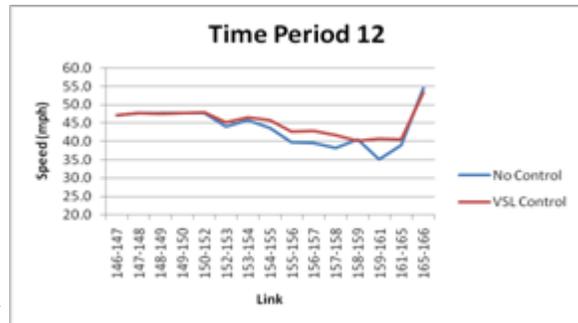
I



J

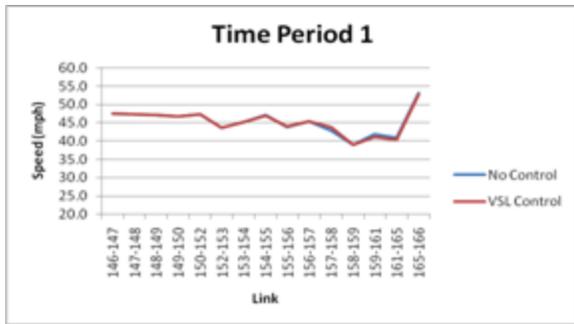


K

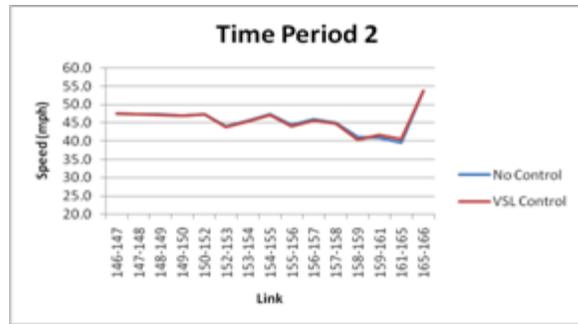


L

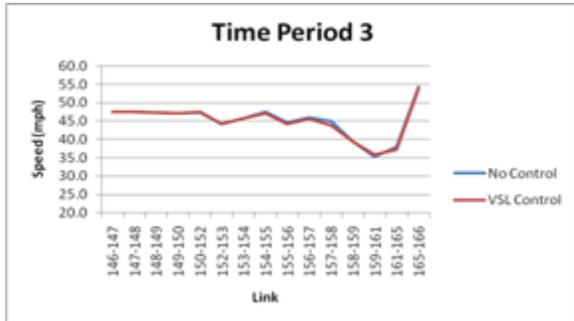
Figure C-8. Continued



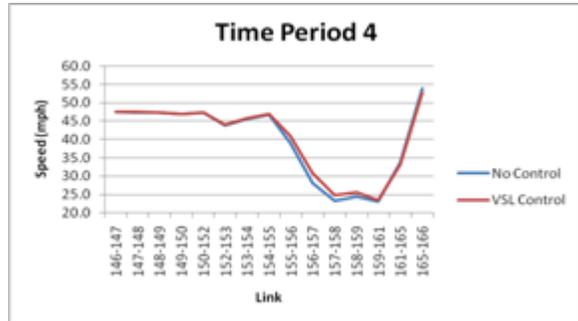
A



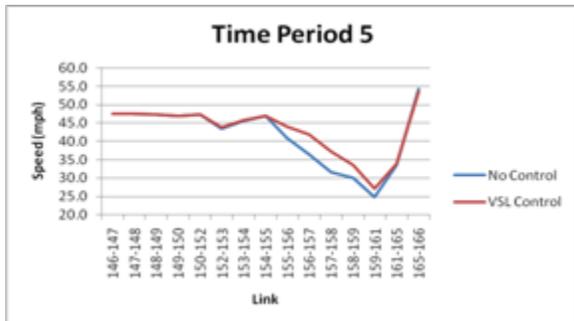
B



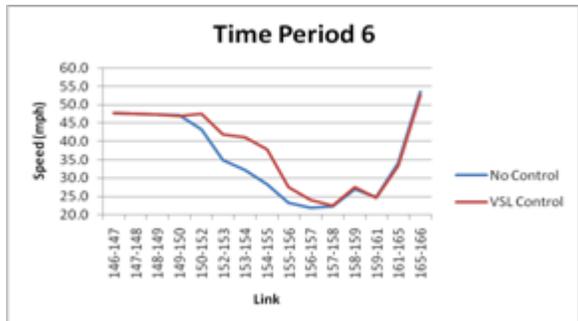
C



D

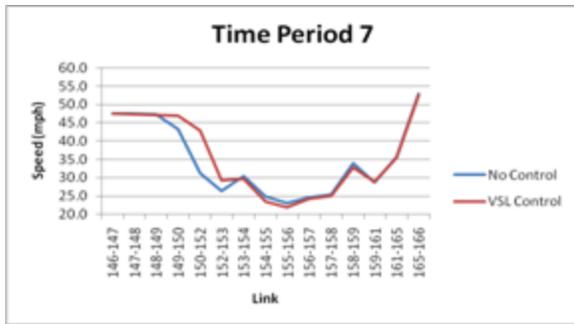


E

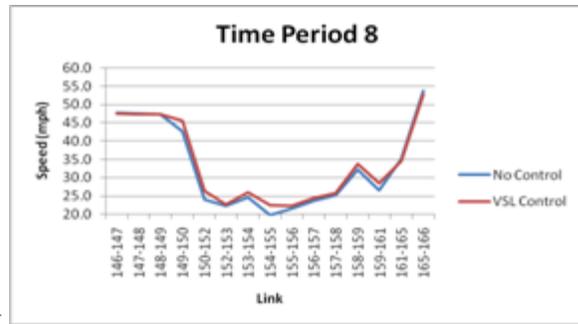


F

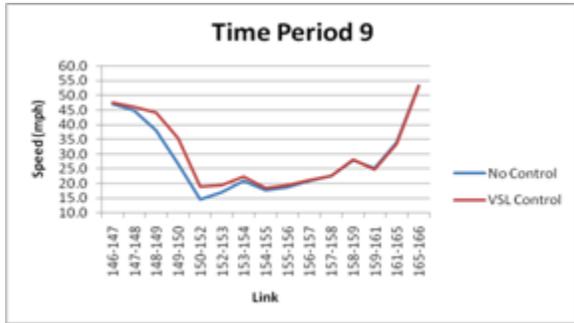
Figure C-9. Speed profile for occupancy-based algorithm using 2 signs spaced 1/2 mile apart (threshold scenario 3) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



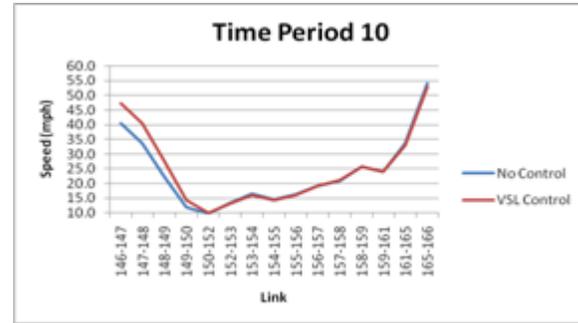
G



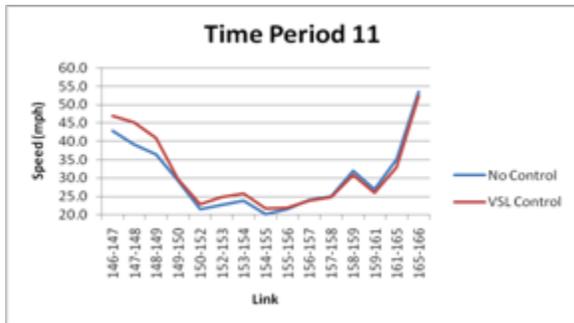
H



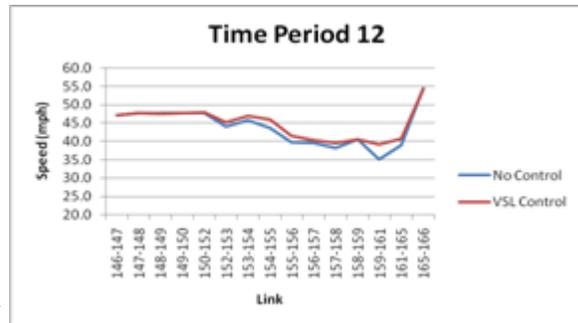
I



J

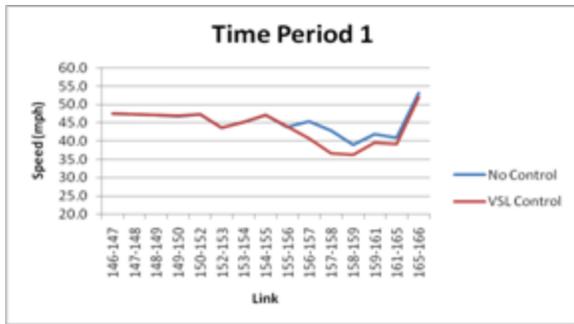


K

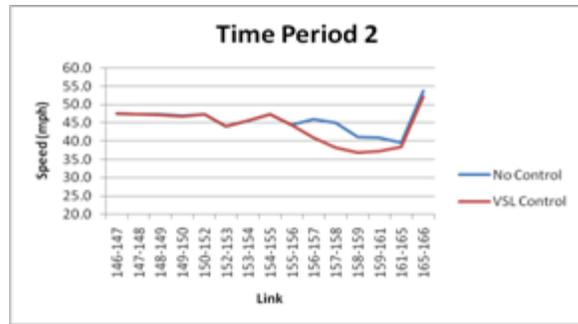


L

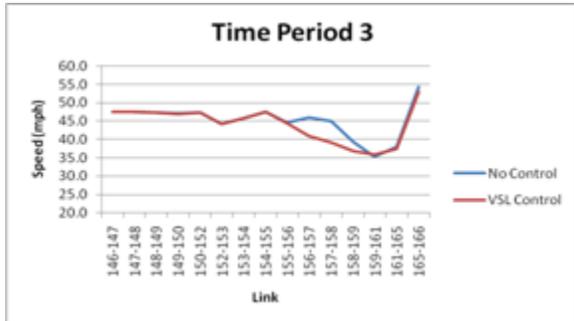
Figure C-9. Continued



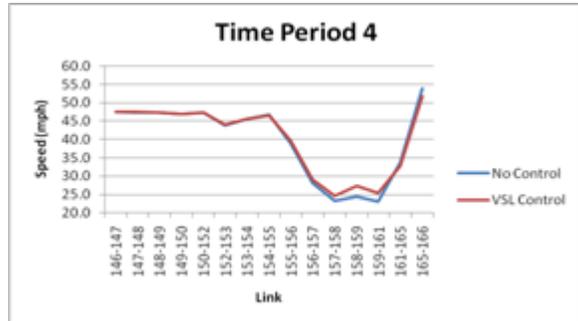
A



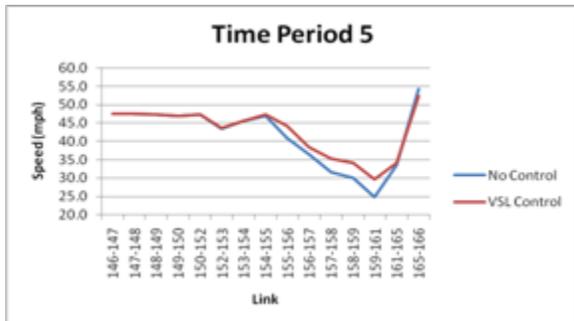
B



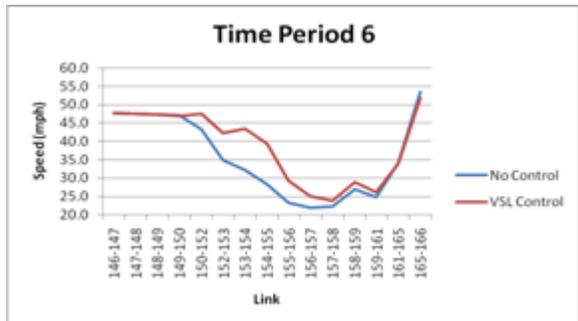
C



D

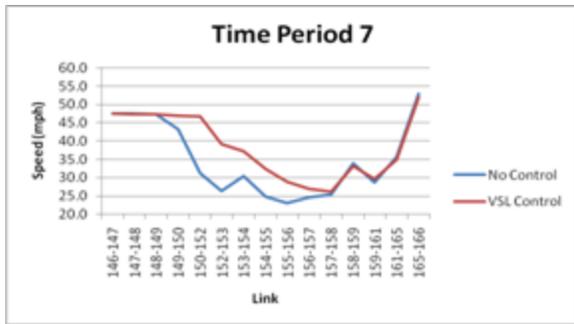


E

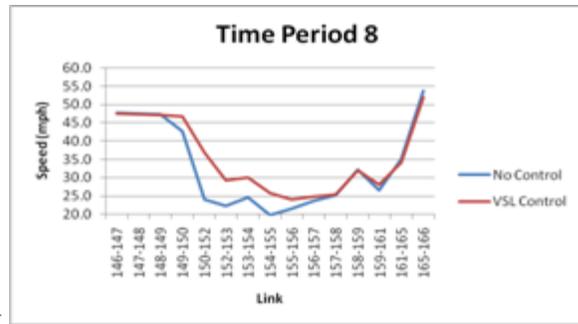


F

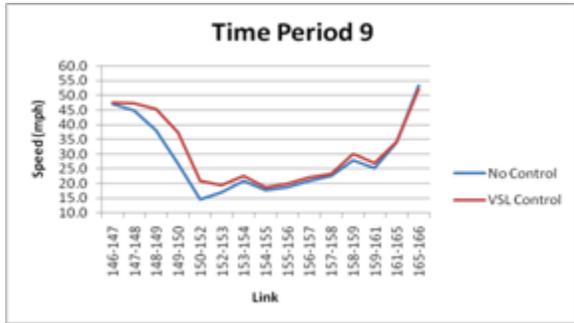
Figure C-10. Speed profile for occupancy-based algorithm using 2 signs spaced 1 mile apart (threshold scenario 1) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



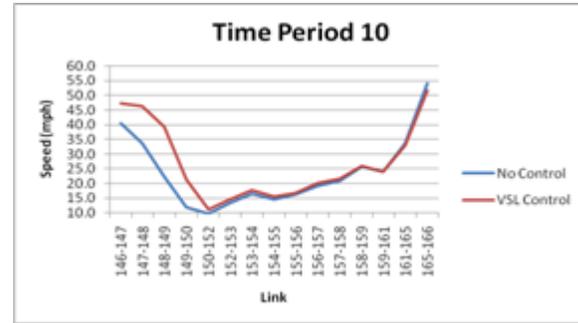
G



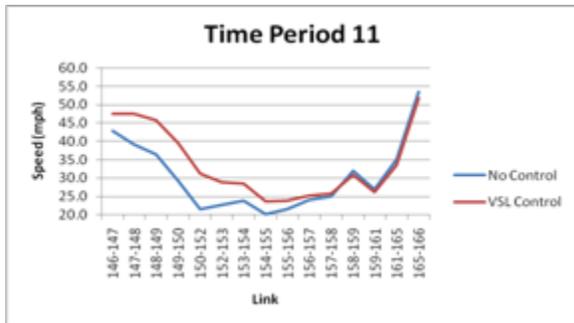
H



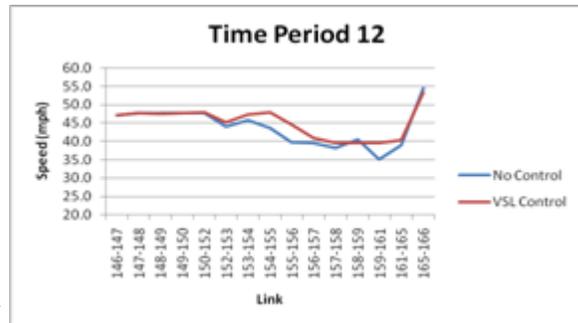
I



J



K



L

Figure C-10. Continued

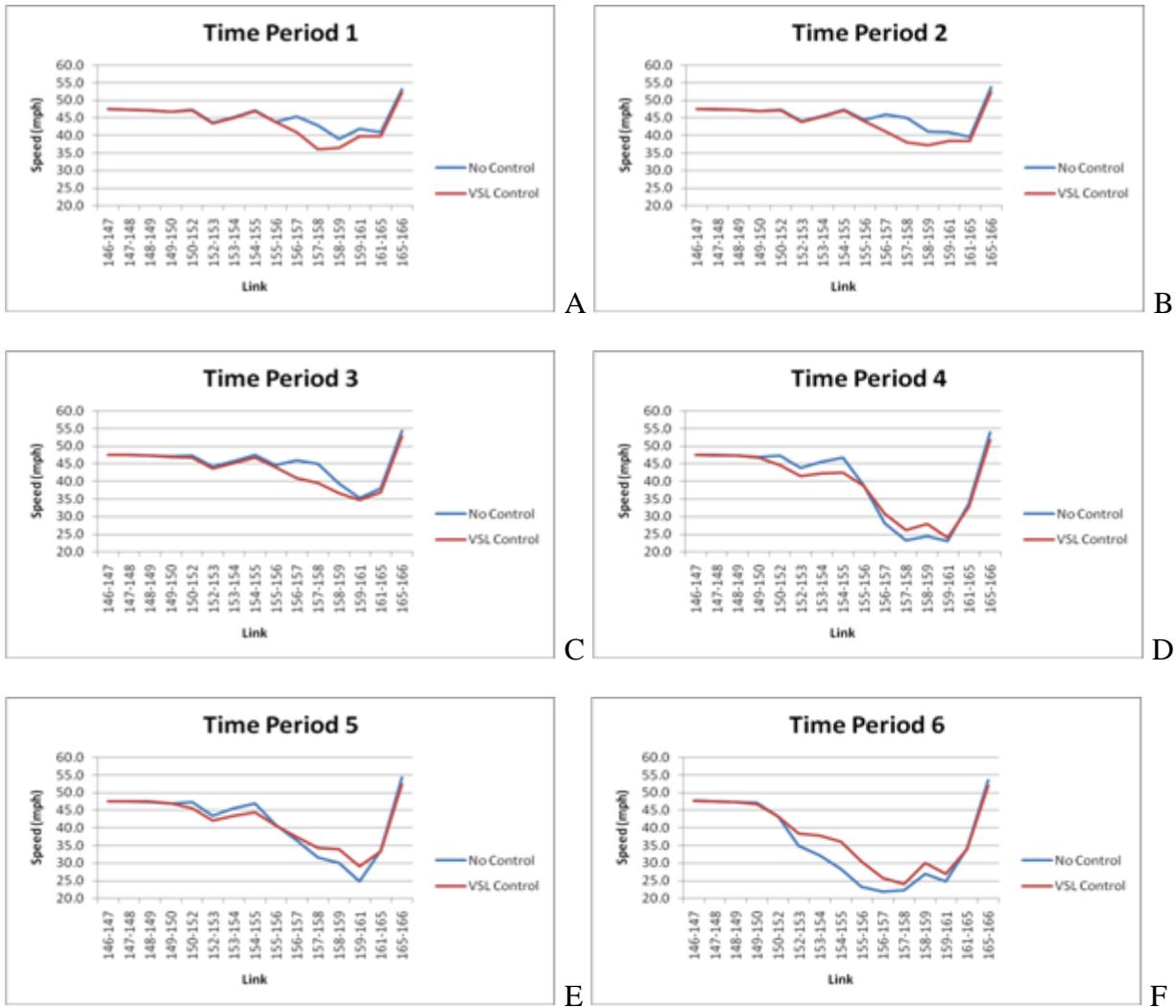
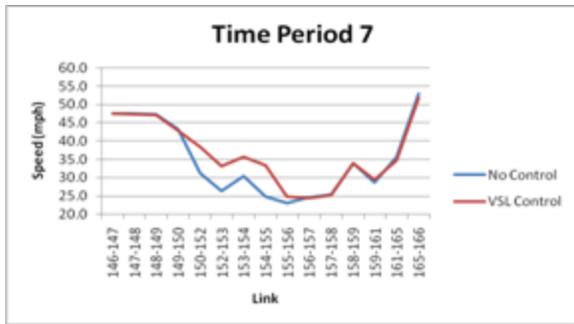
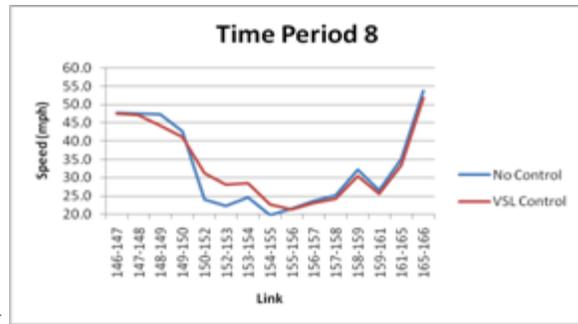


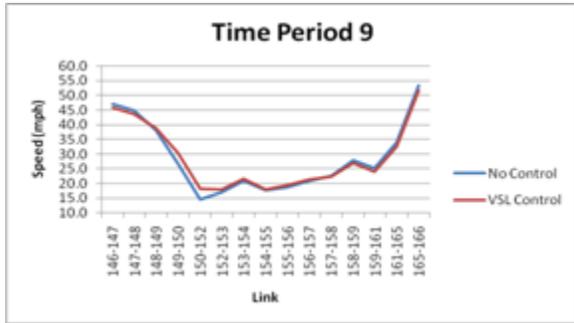
Figure C-11. Speed profile for occupancy-based algorithm using 2 signs spaced 1 mile apart (threshold scenario 2) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



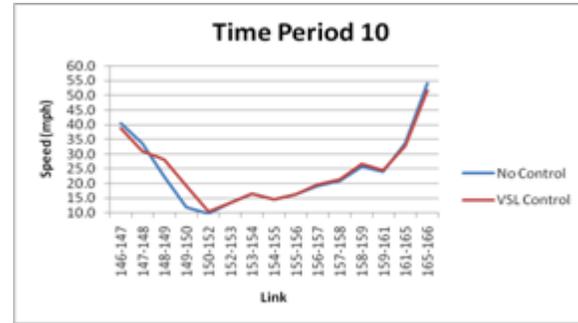
G



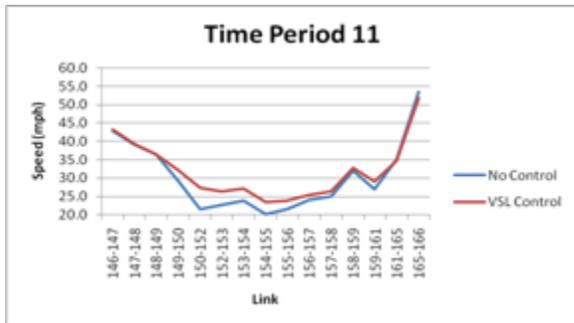
H



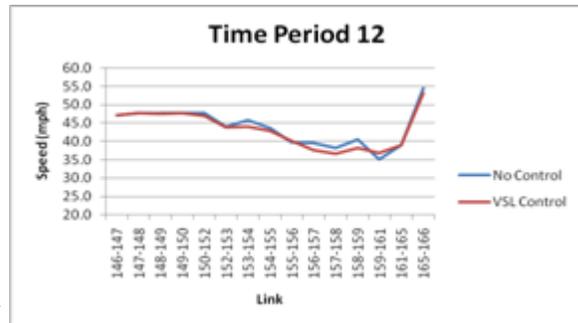
I



J

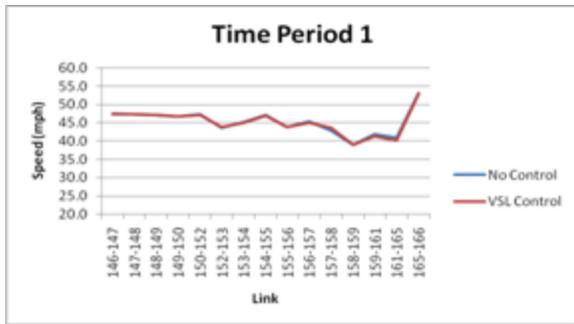


K

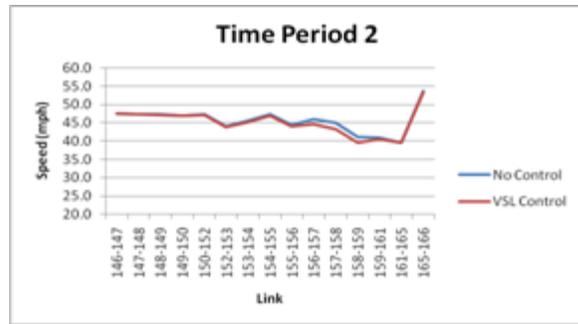


L

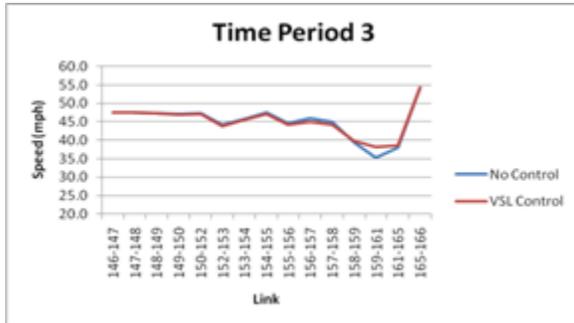
Figure C-11. Continued



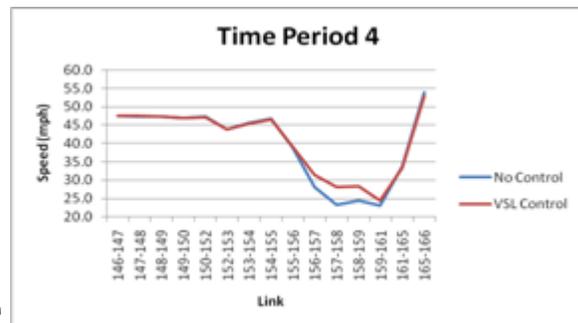
A



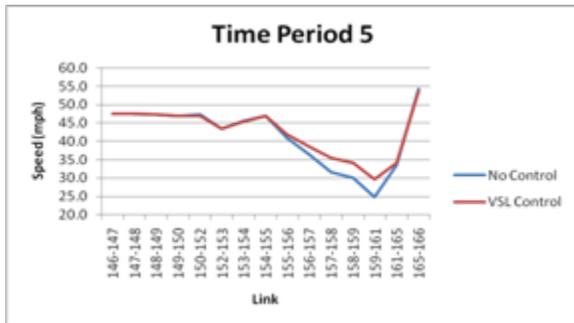
B



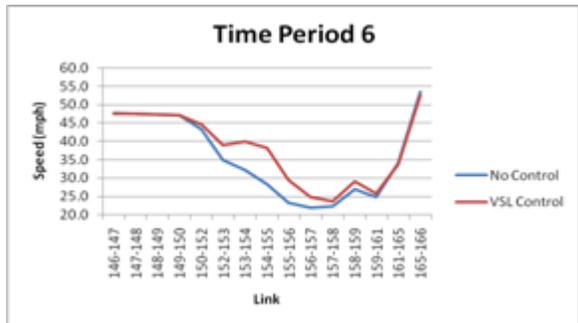
C



D

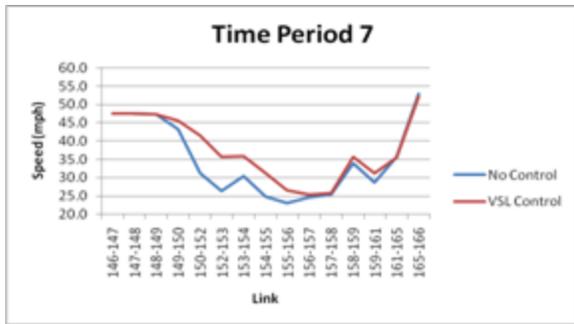


E

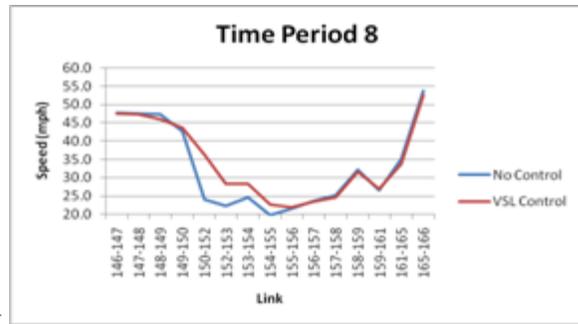


F

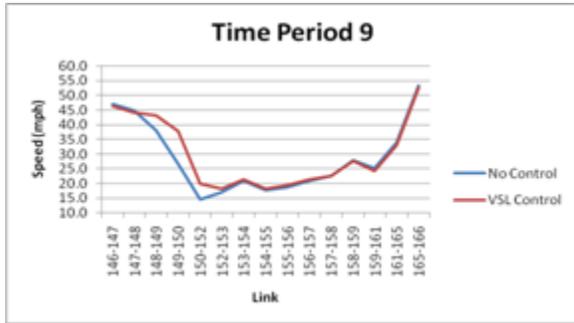
Figure C-12. Speed profile for occupancy-based algorithm using 2 signs spaced 1 mile apart (threshold scenario 3) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



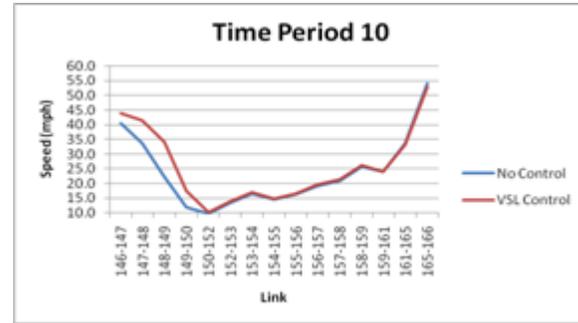
G



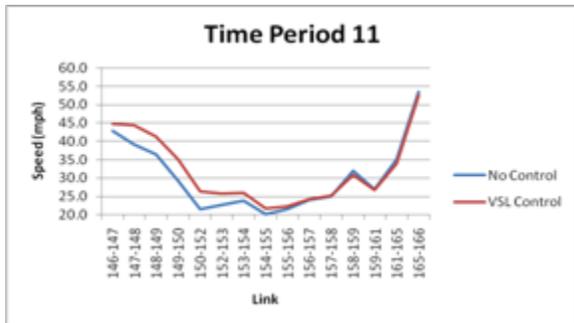
H



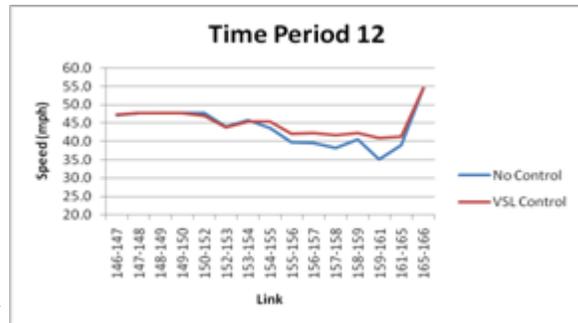
I



J

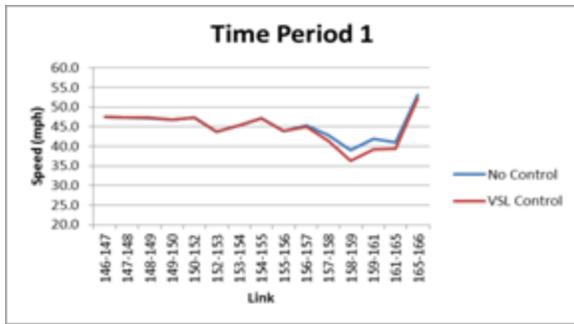


K

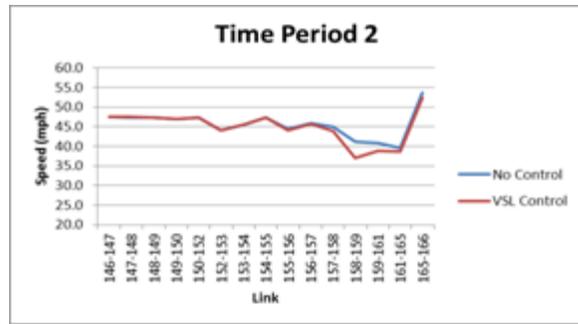


L

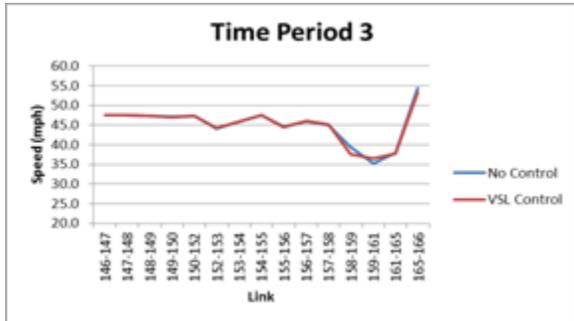
Figure C-12. Continued



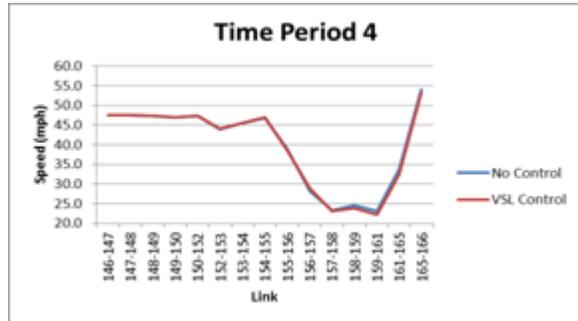
A



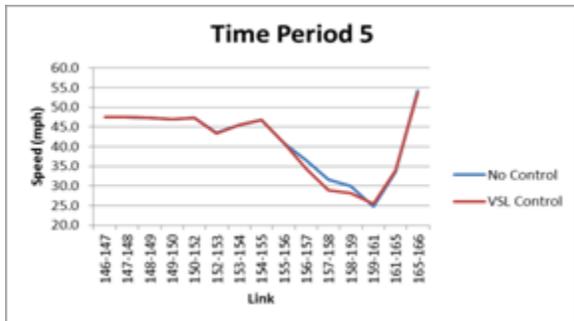
B



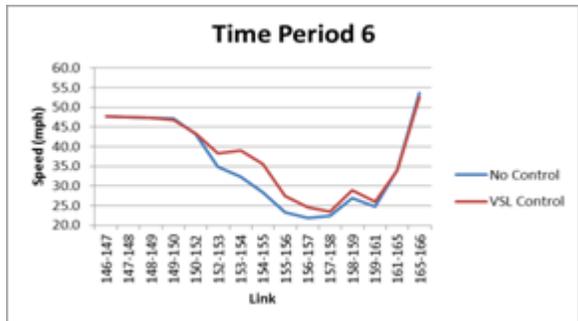
C



D

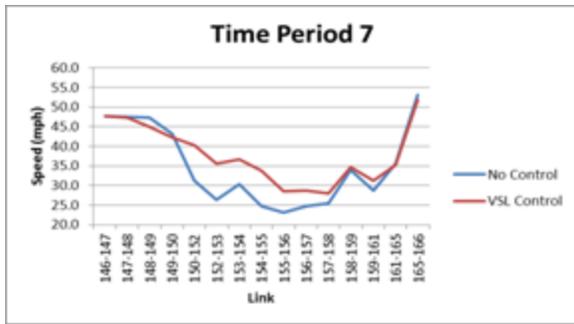


E

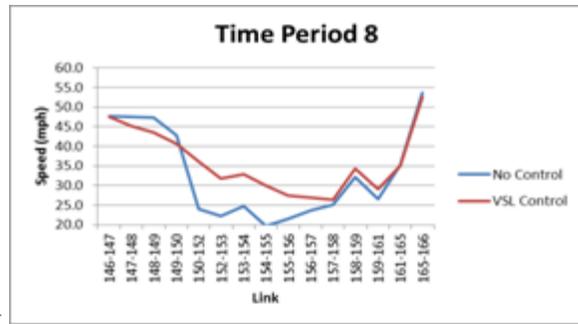


F

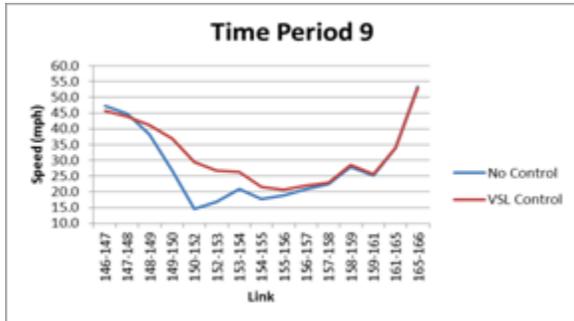
Figure C-13. Speed profile for volume-based algorithm using 1 sign spaced 1/2 mile from the bottleneck source (threshold scenario 1) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time Period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



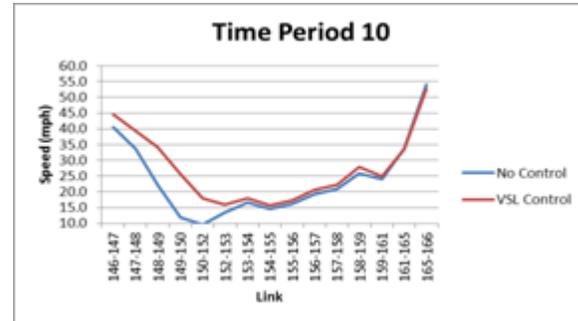
G



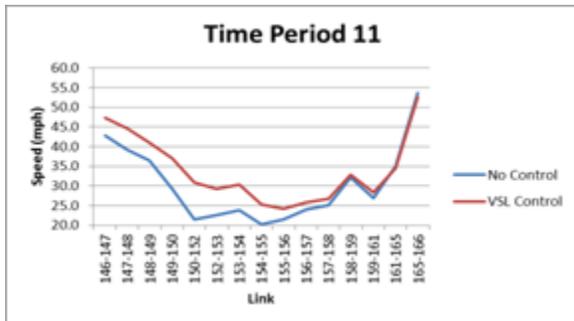
H



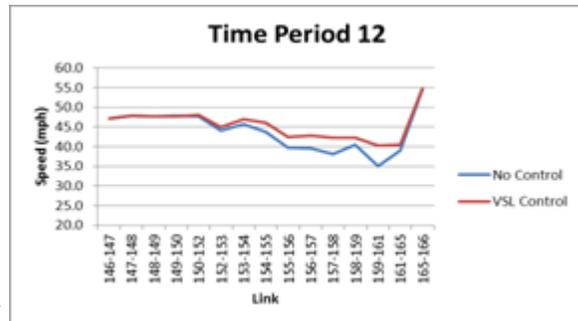
I



J

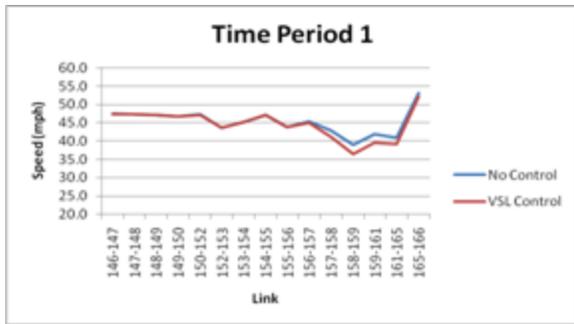


K

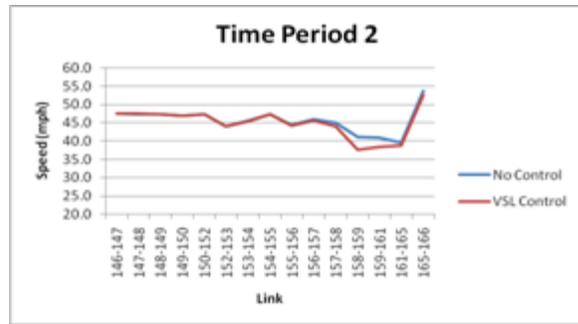


L

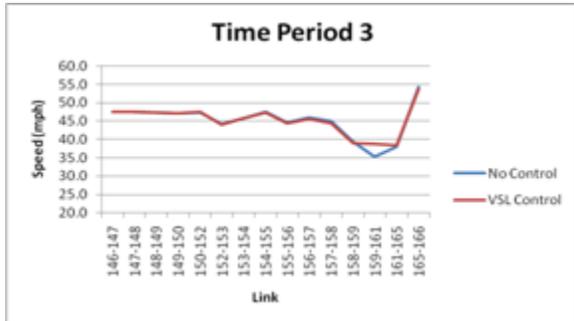
Figure C-13. Continued



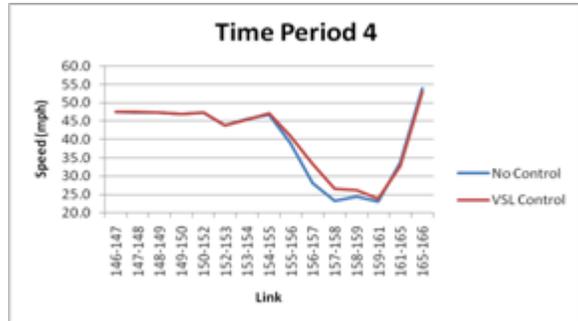
A



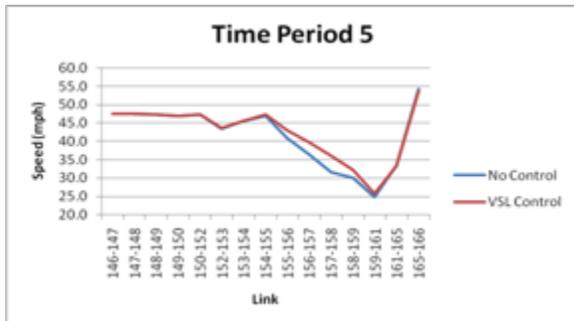
B



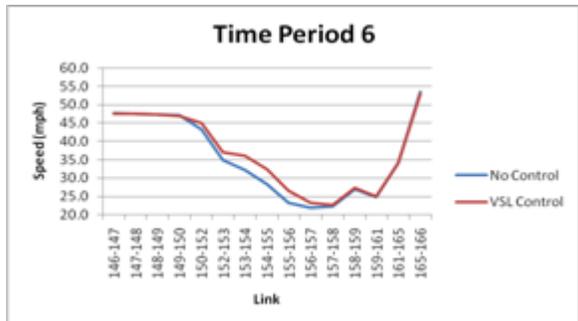
C



D

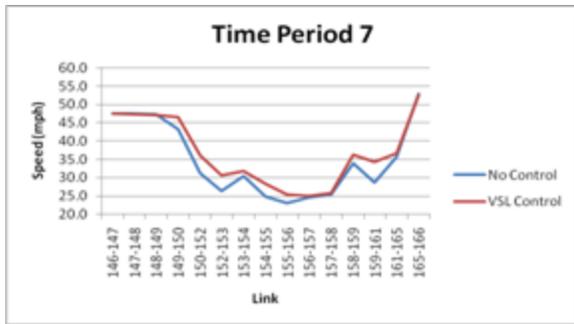


E

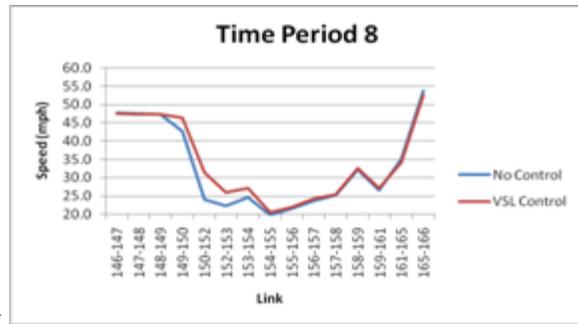


F

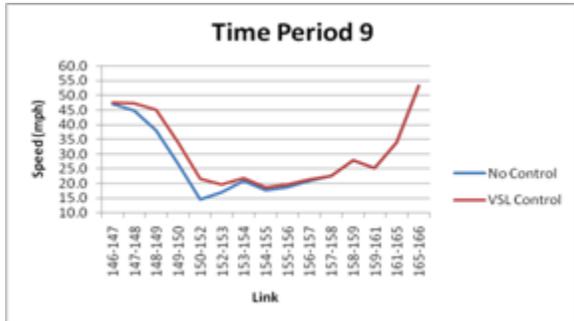
Figure C-14. Speed profile for volume-based algorithm using 1 sign spaced 1/2 mile from the bottleneck source (threshold scenario 2) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time Period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



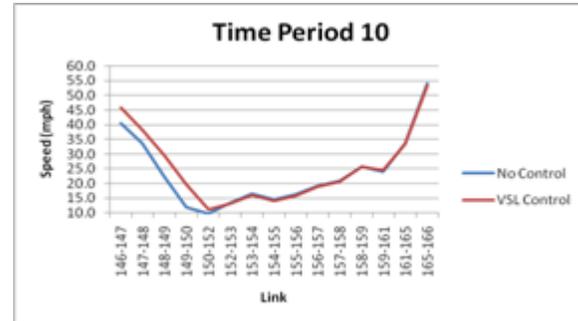
G



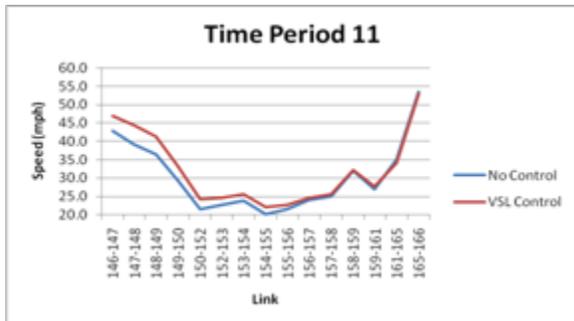
H



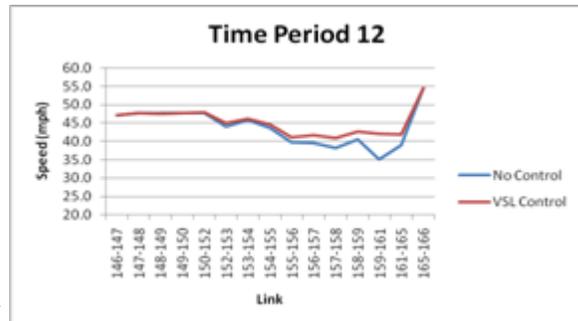
I



J

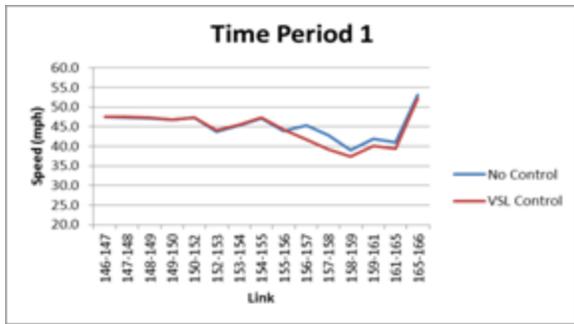


K

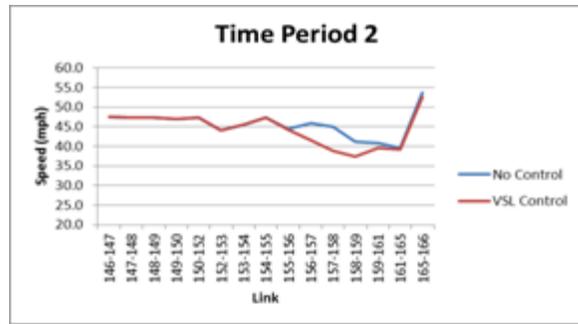


L

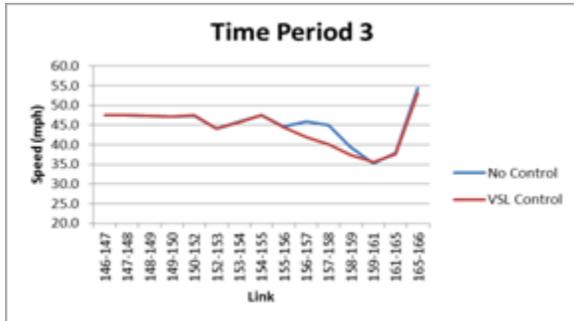
Figure C-14. Continued



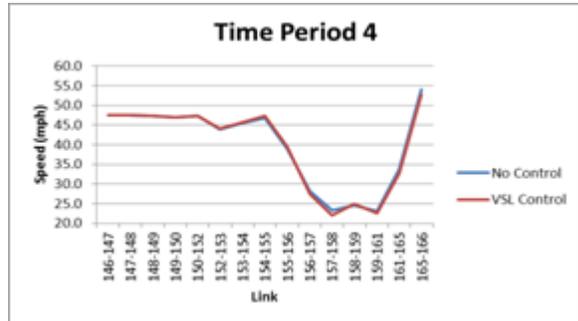
A



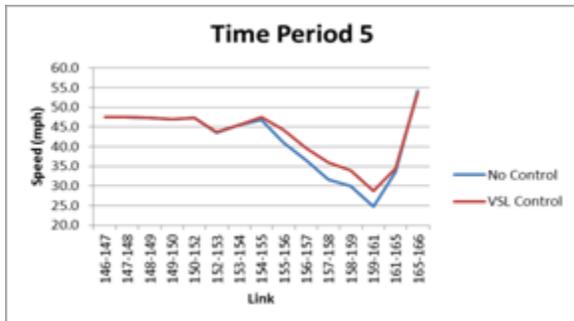
B



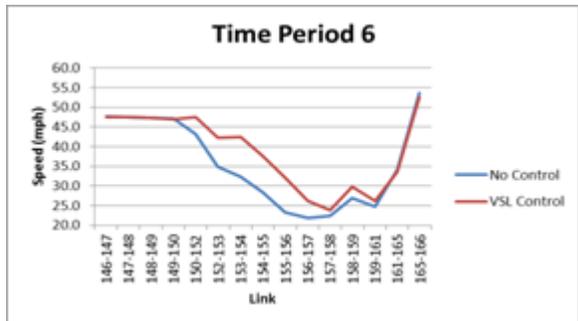
C



D

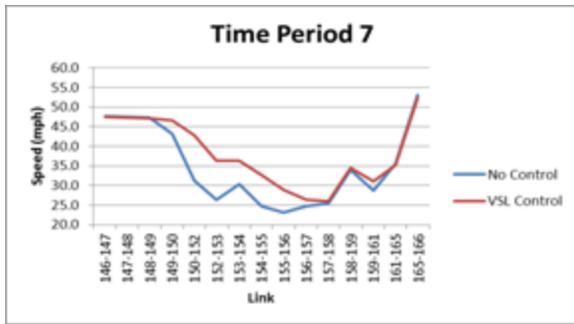


E

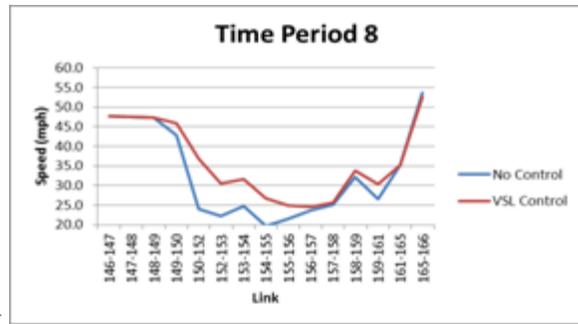


F

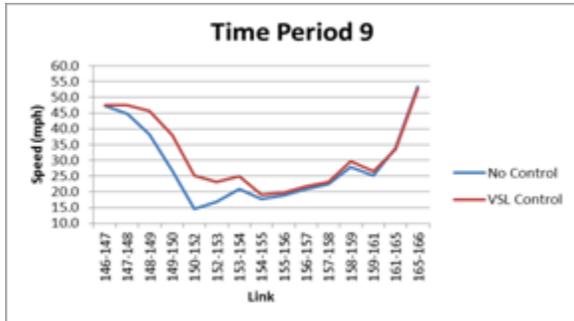
Figure C-15. Speed profile for volume-based algorithm using 1 sign spaced 1 mile from the bottleneck source (threshold scenario 1) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time Period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



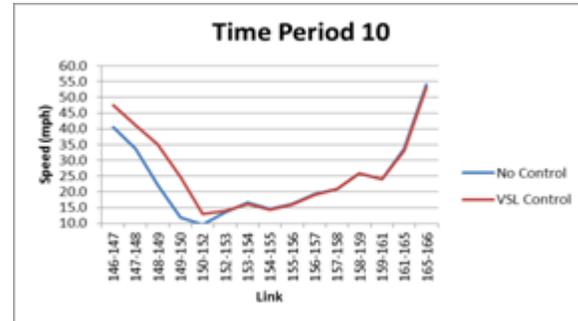
G



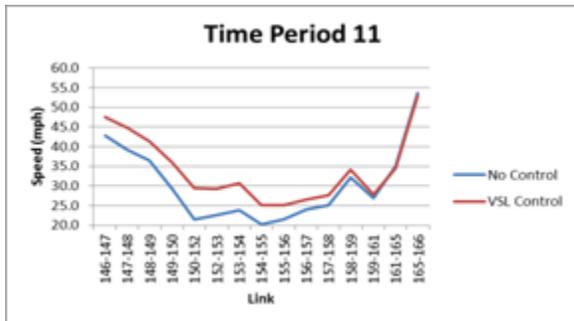
H



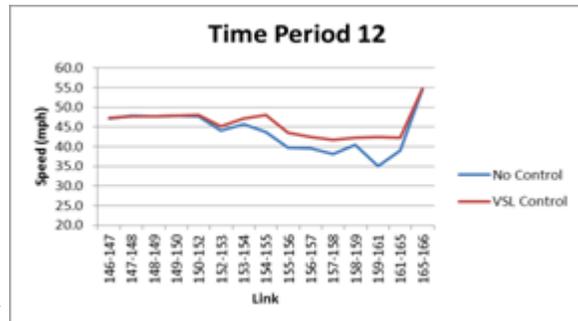
I



J



K



L

Figure C-15. Continued

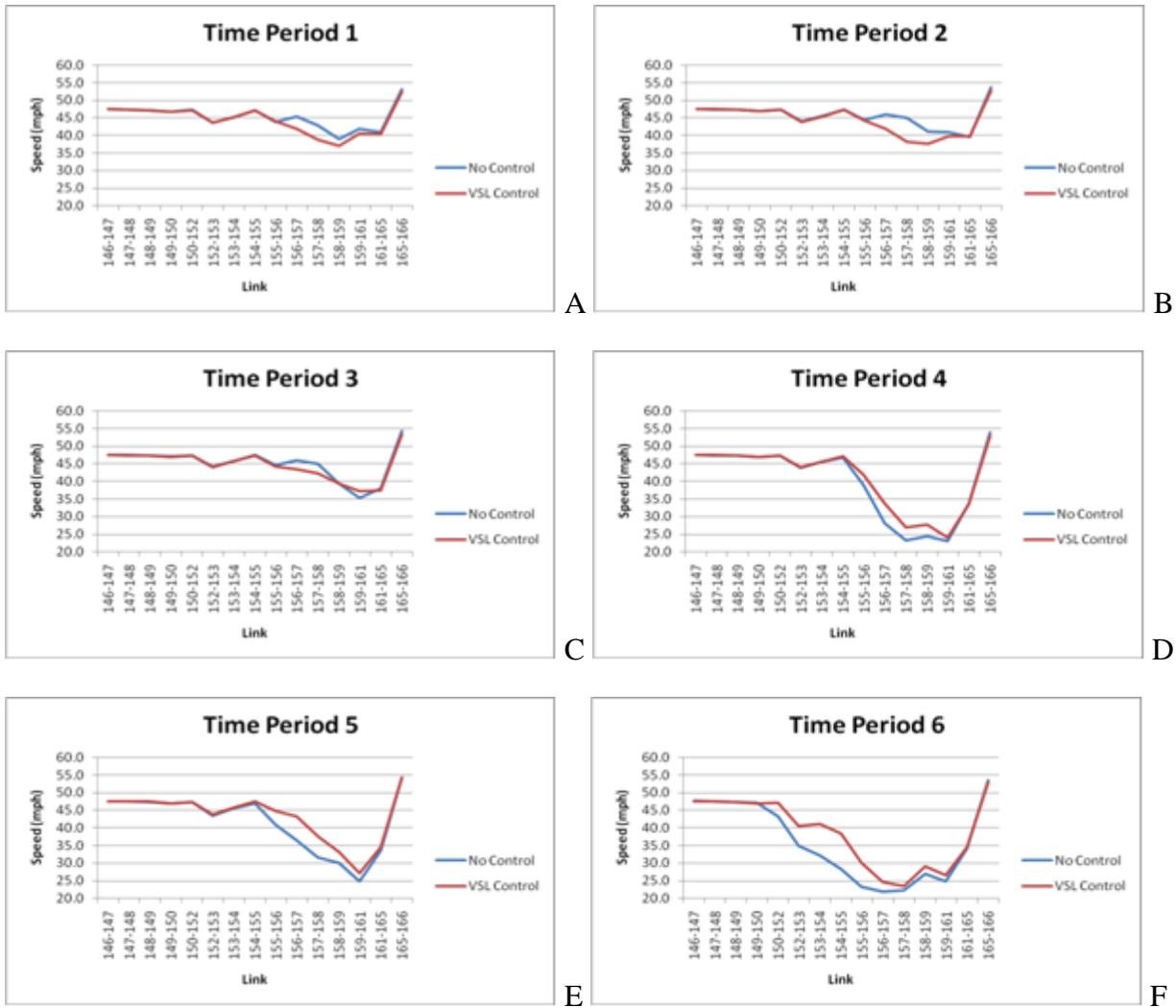
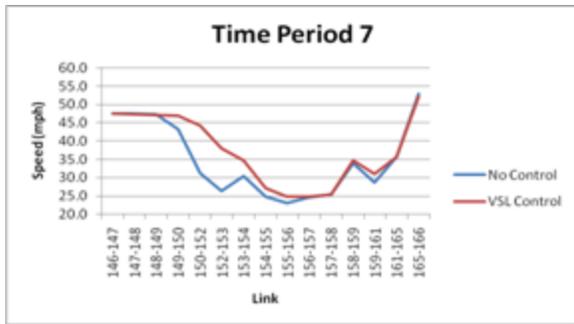
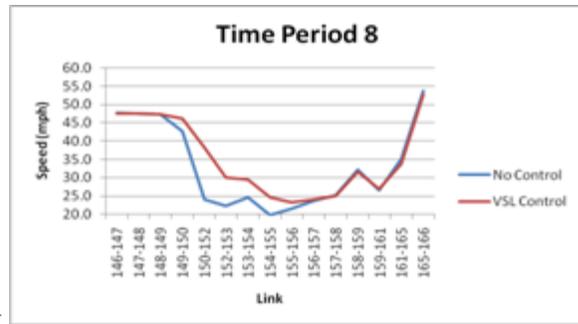


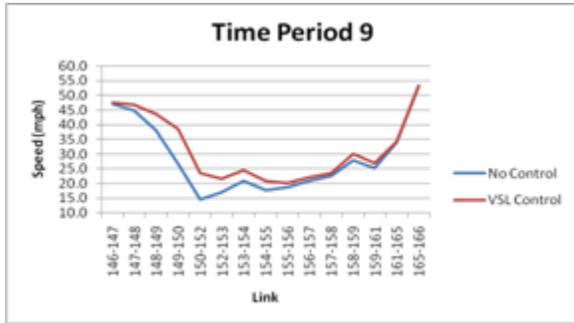
Figure C-16. Speed profile for volume-based algorithm using 1 sign spaced 1 mile from the bottleneck source (threshold scenario 2) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time Period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



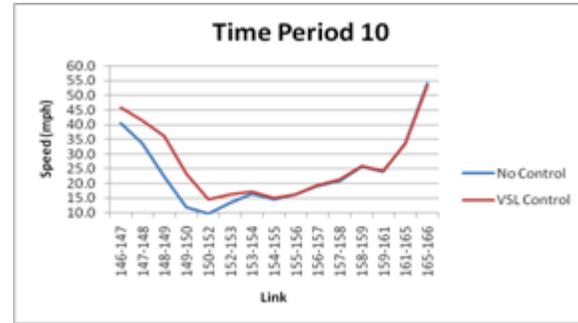
G



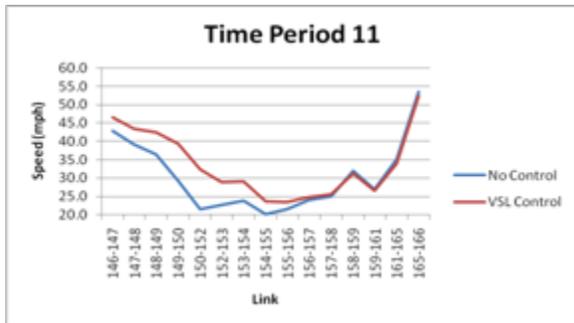
H



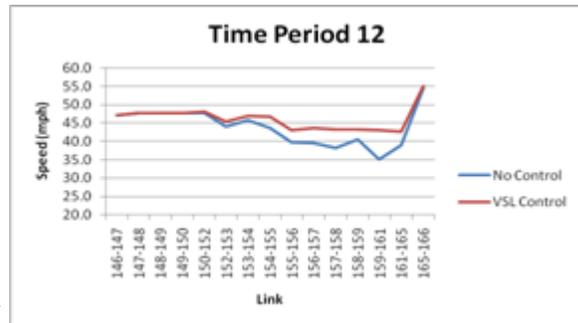
I



J

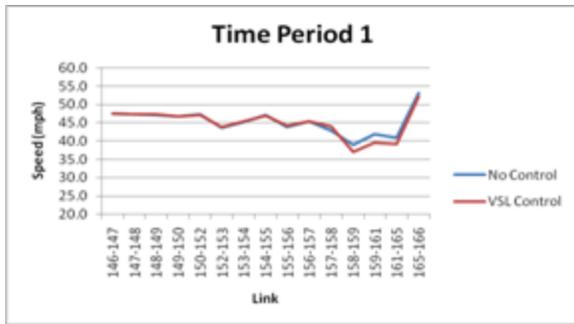


K

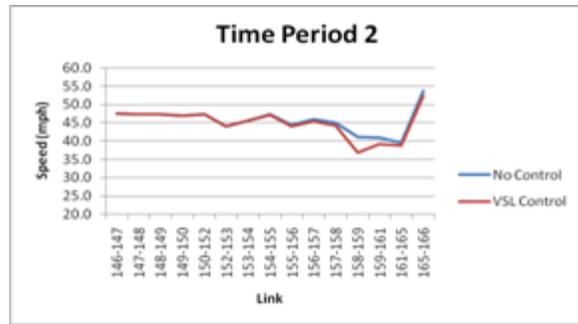


L

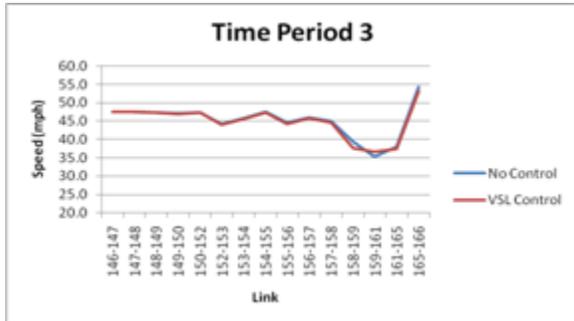
Figure C-16. Continued



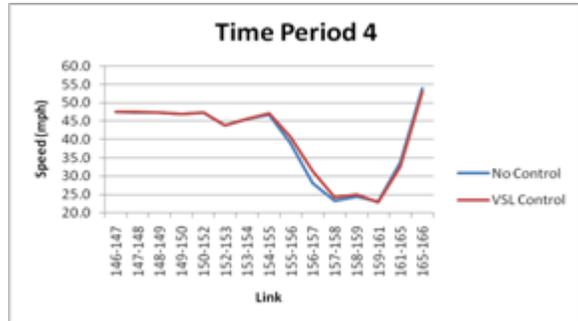
A



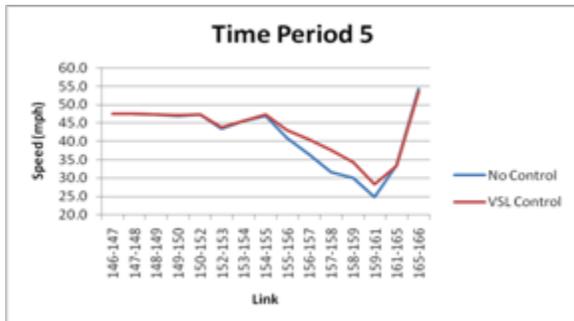
B



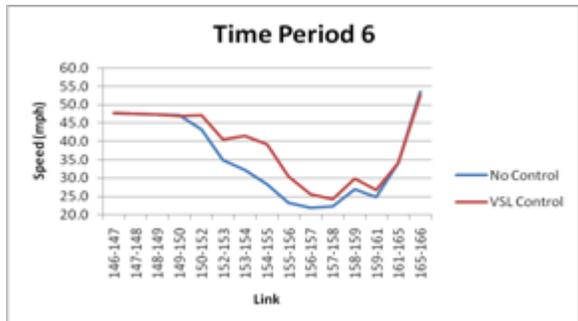
C



D

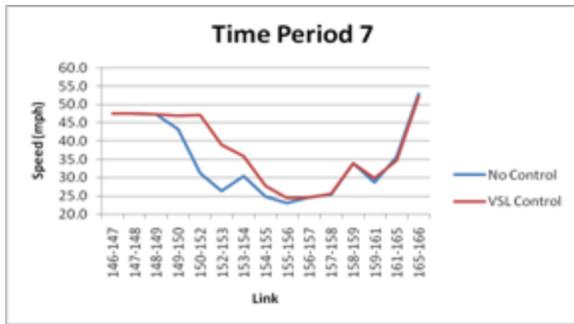


E

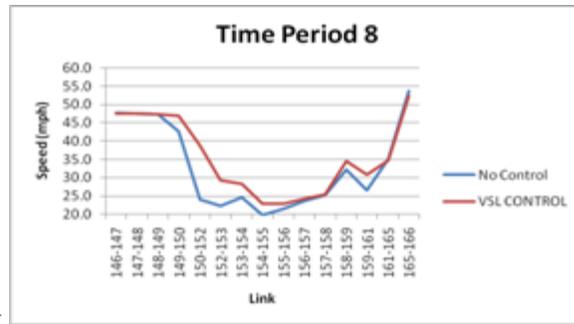


F

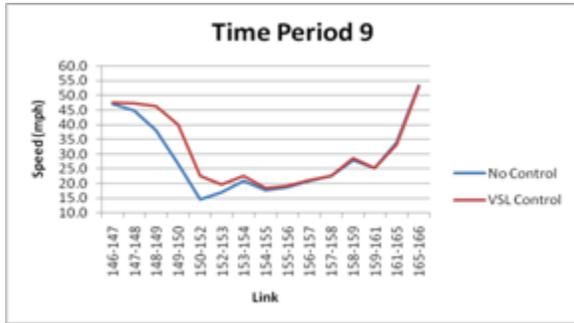
Figure C-17. Speed profile for volume-based algorithm using 2 signs spaced 1/2 mile apart (threshold scenario 1) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



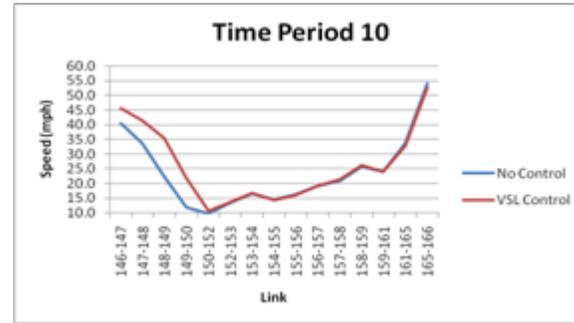
G



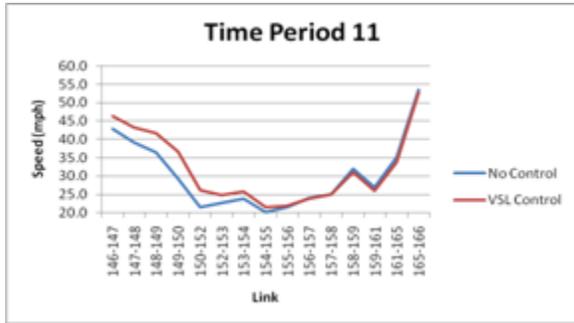
H



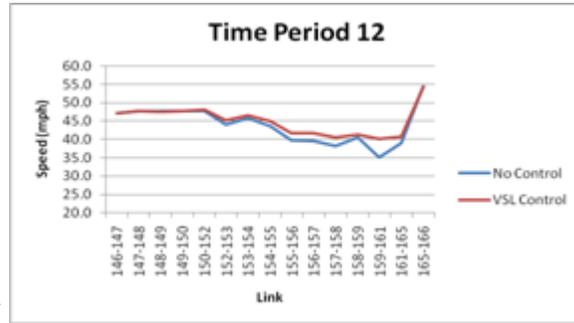
I



J



K



L

Figure C-17. Continued

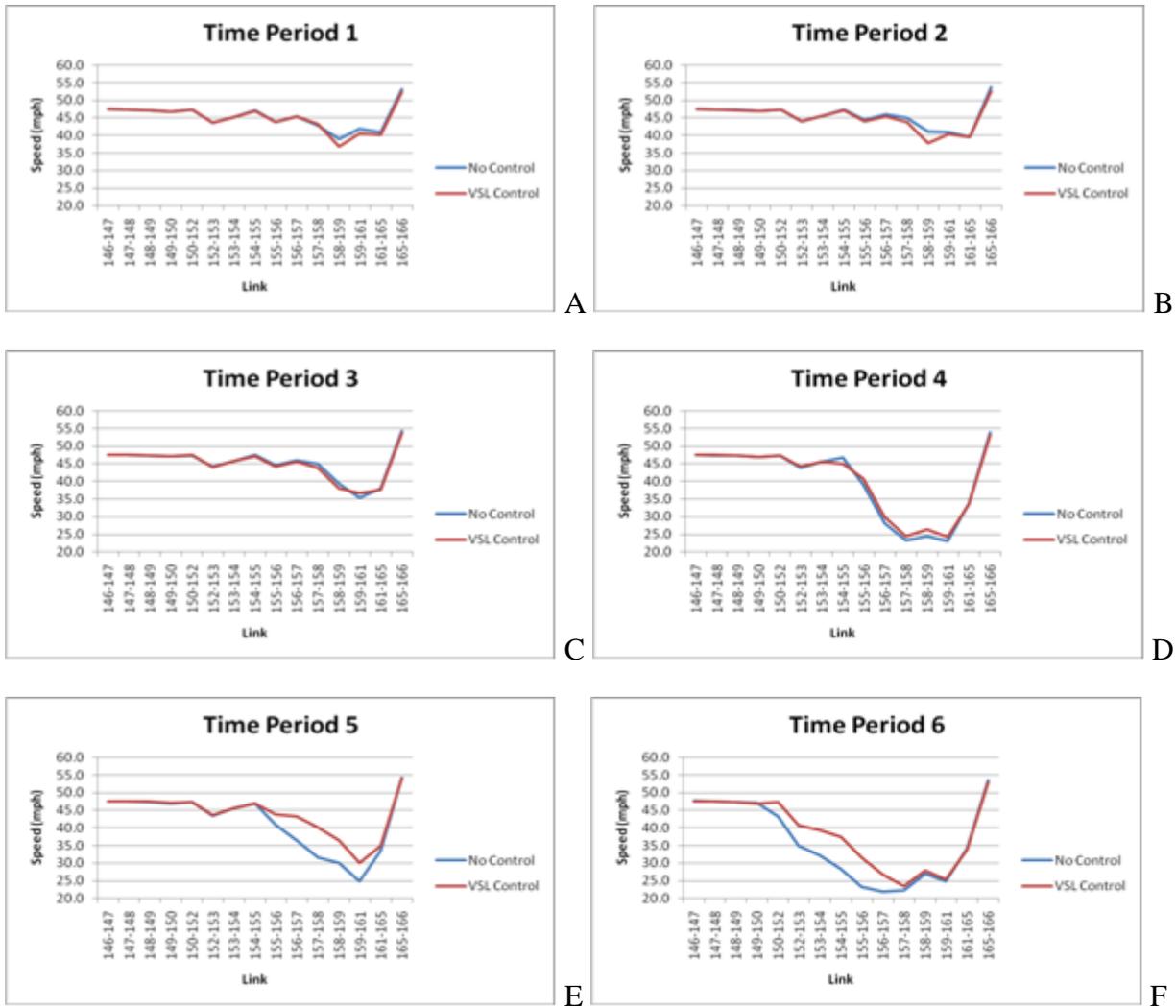
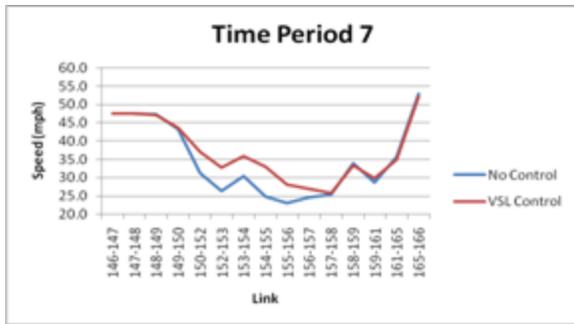
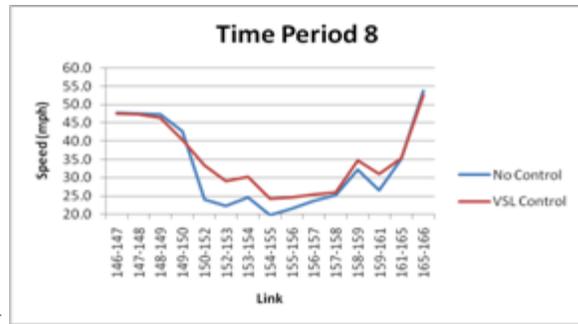


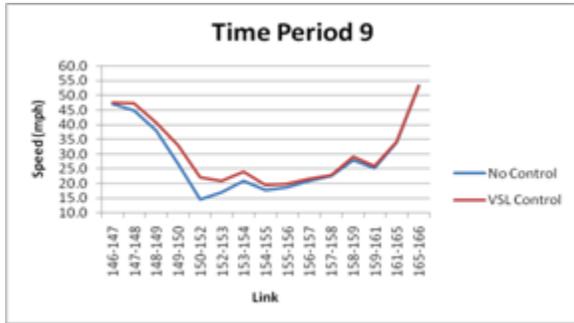
Figure C-18. Speed profile for volume-based algorithm using 2 sign spaced 1/2 mile apart (threshold scenario 2) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



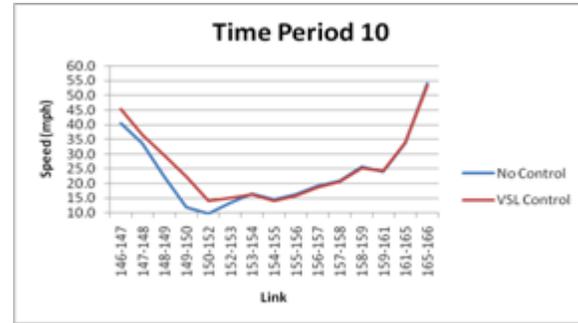
G



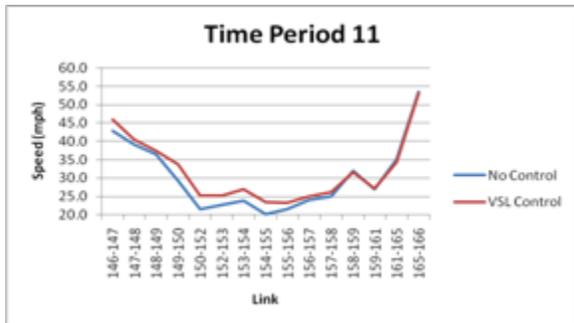
H



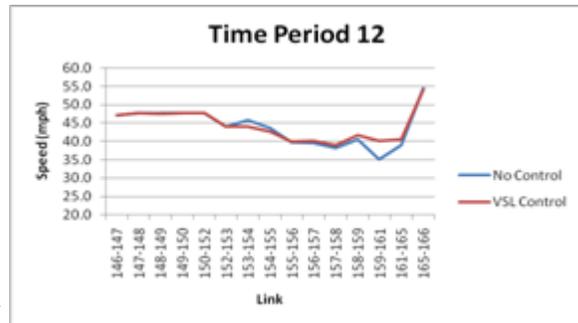
I



J



K



L

Figure C-18. Continued

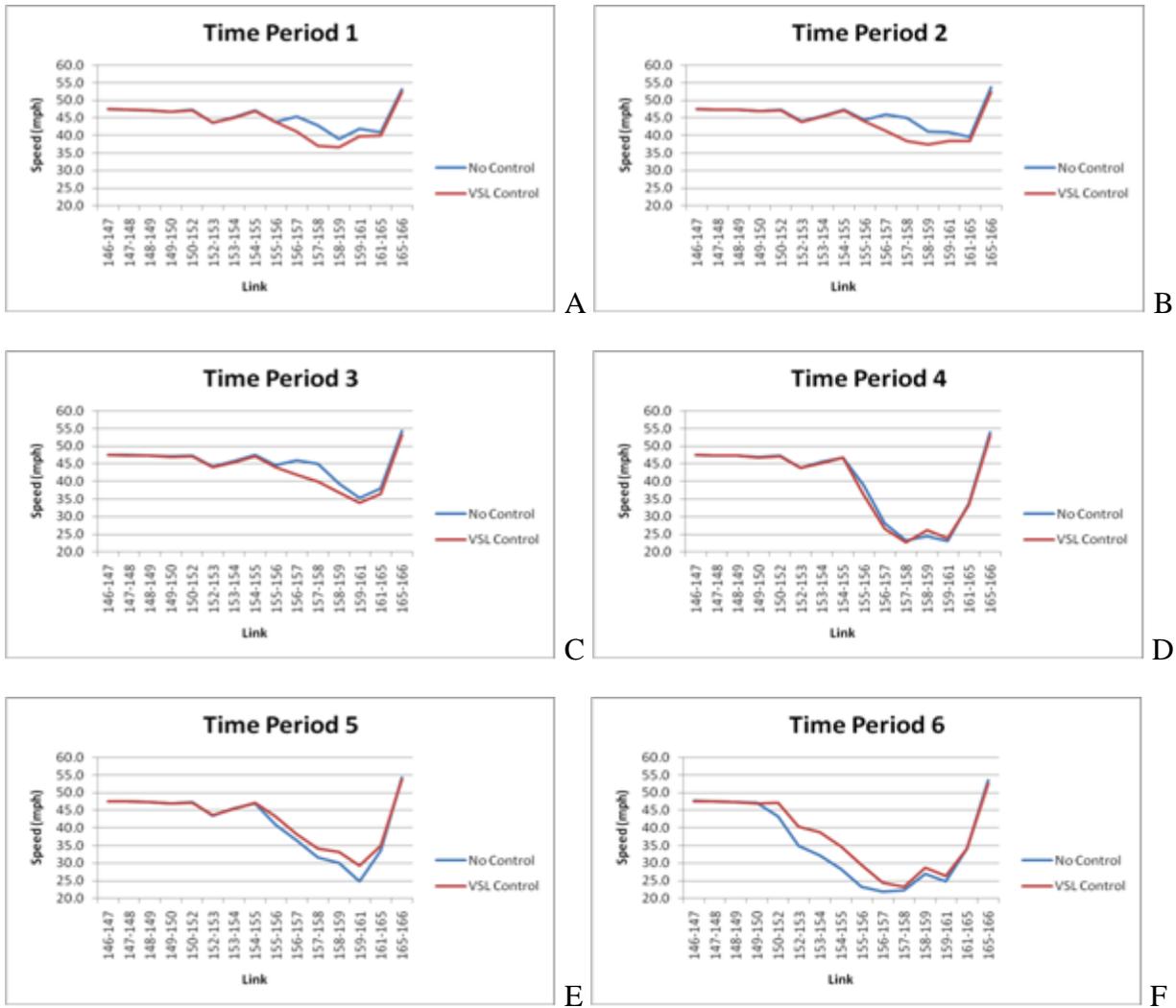
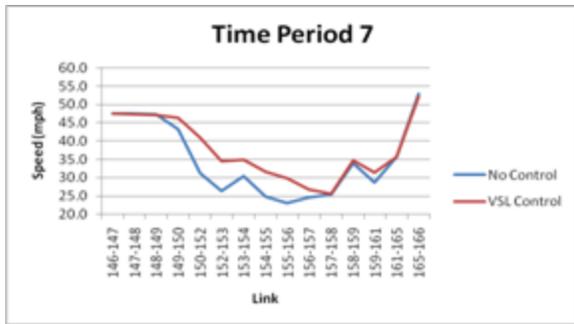
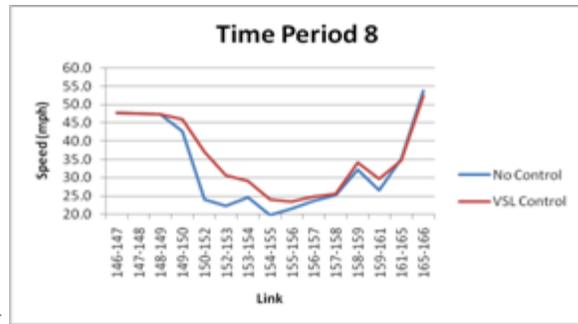


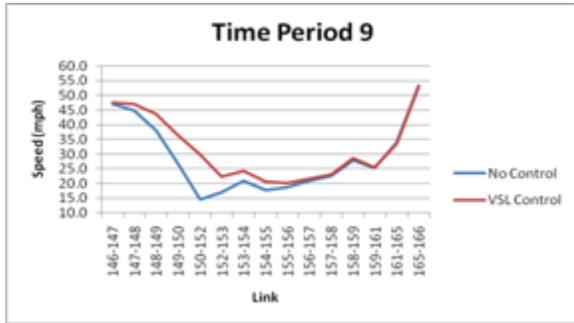
Figure C-19. Speed profile for volume-based algorithm using 2 signs spaced 1 mile apart (threshold scenario 1) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



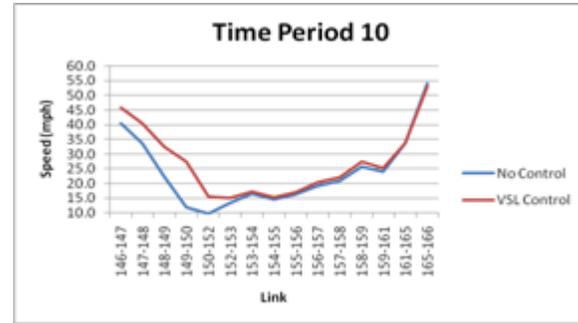
G



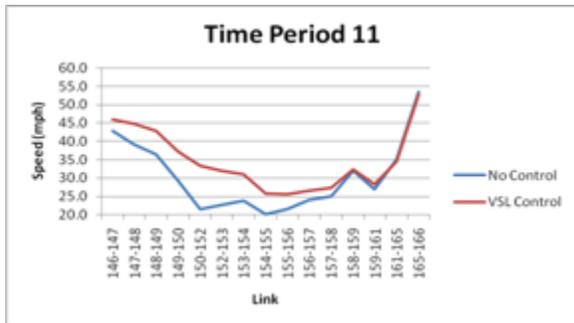
H



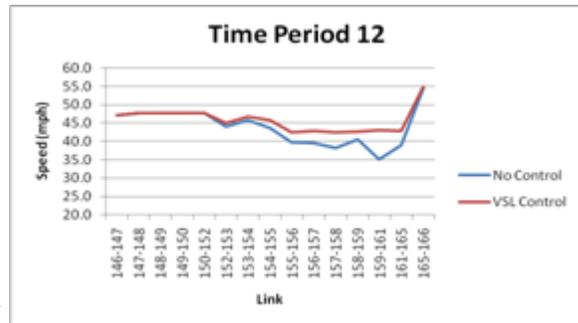
I



J

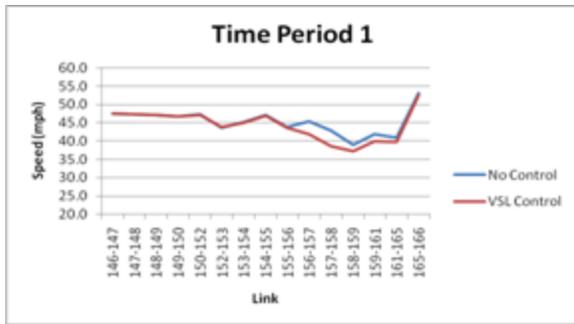


K

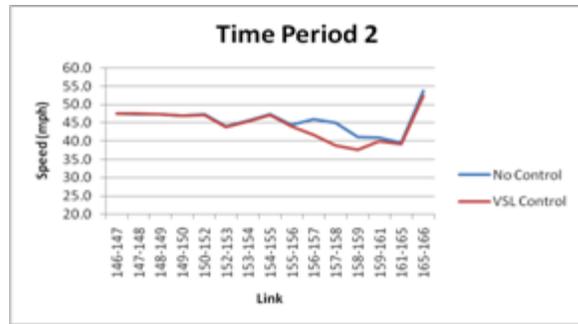


L

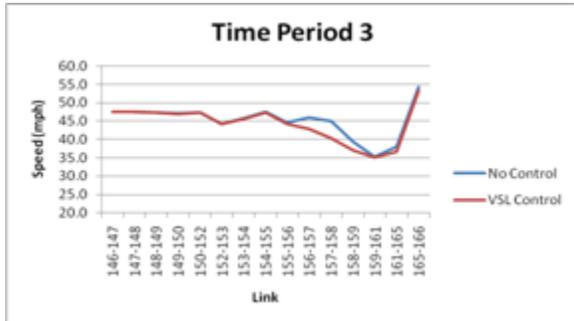
Figure C-19. Continued



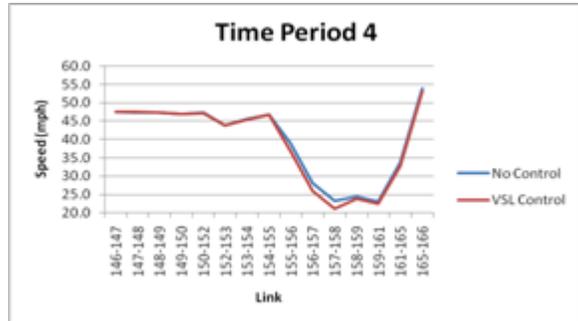
A



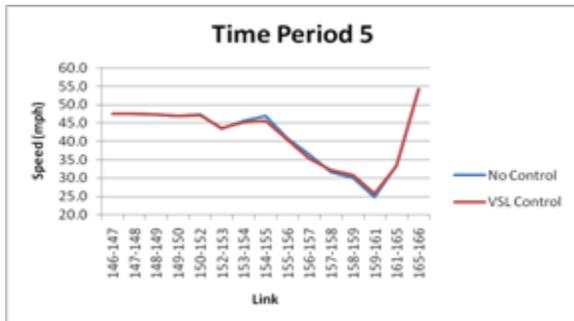
B



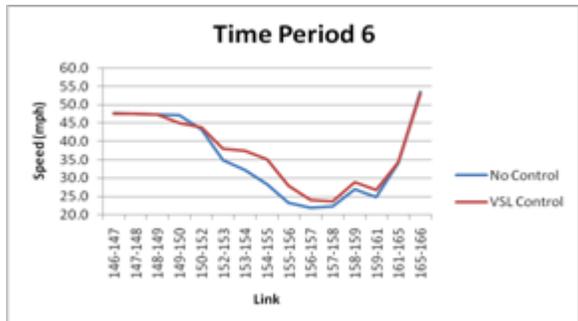
C



D

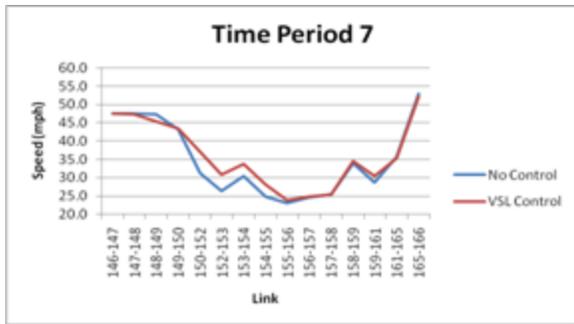


E

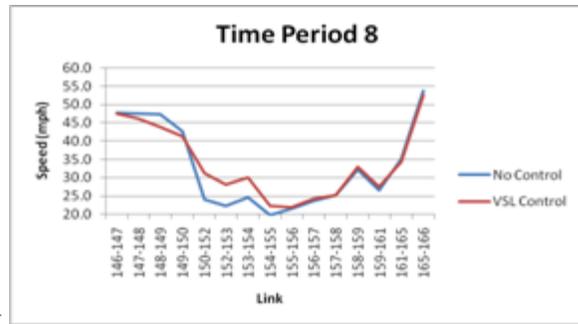


F

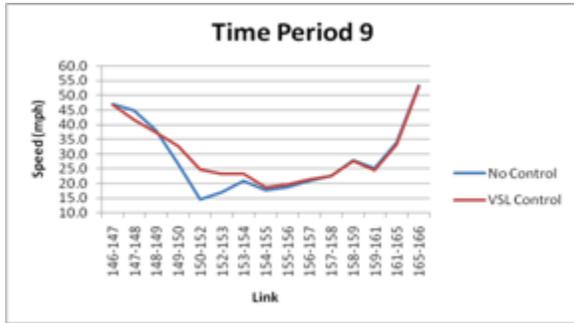
Figure C-20. Speed profile for volume-based algorithm using 2 sign spaced 1 mile apart (threshold scenario 2) compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



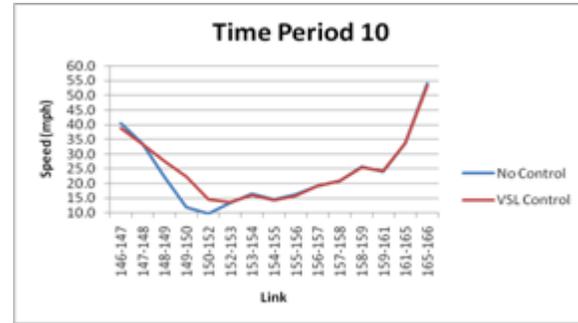
G



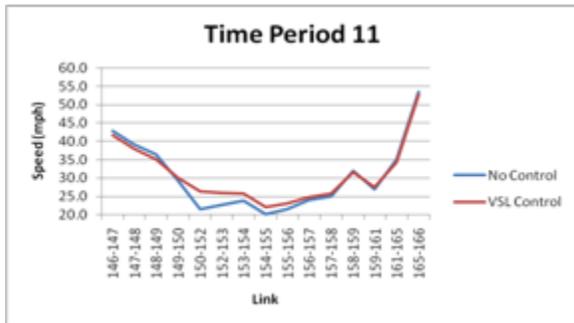
H



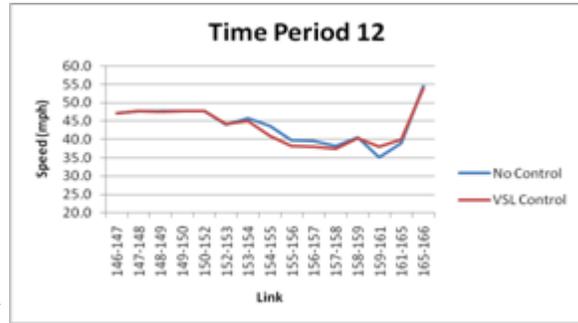
I



J

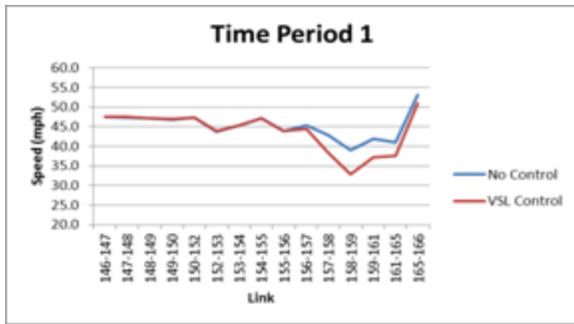


K

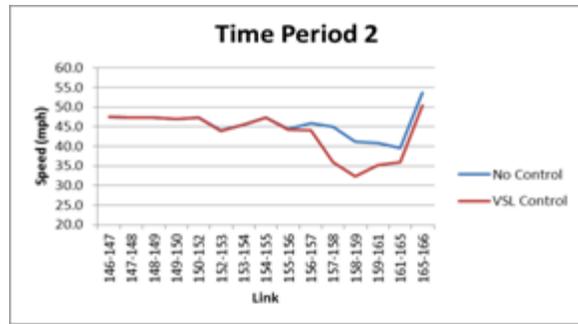


L

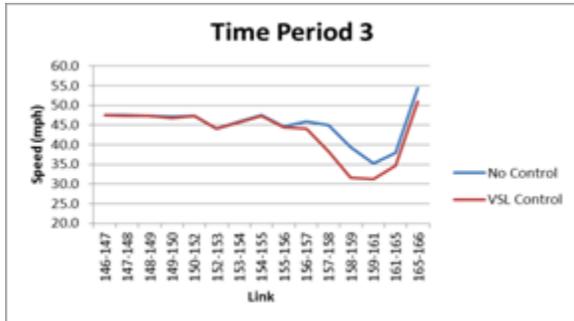
Figure C-20. Continued



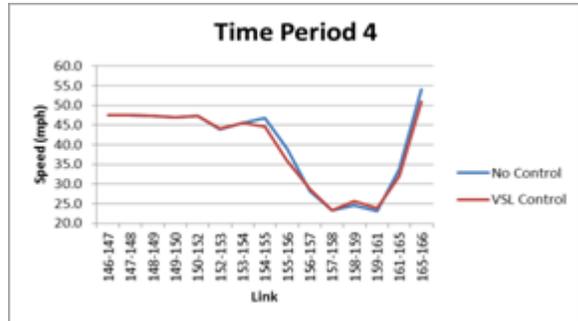
A



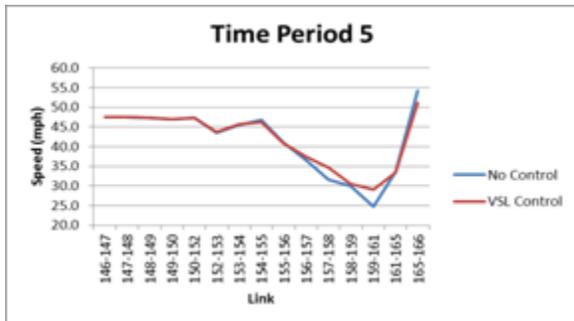
B



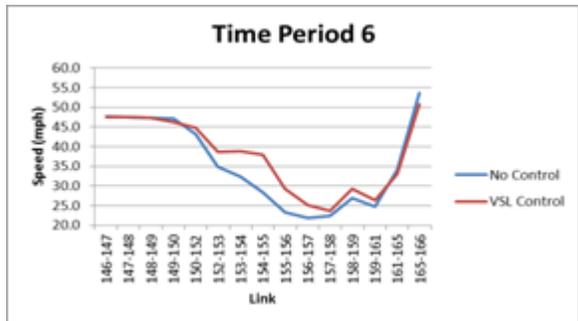
C



D

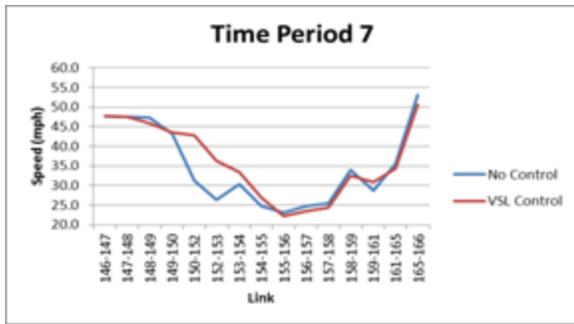


E

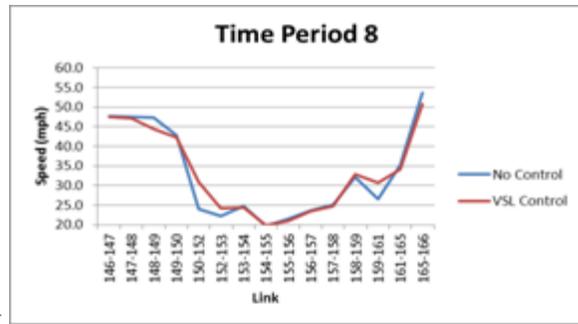


F

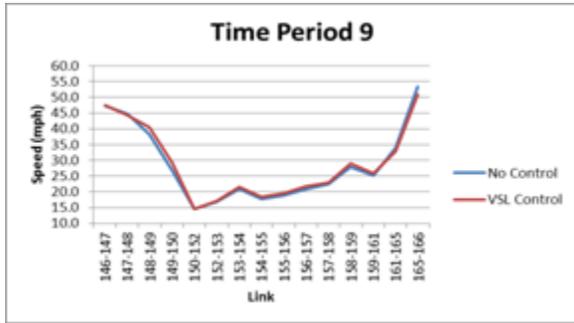
Figure C-21. Speed profile for multiple parameter-based algorithm using 1 sign spaced 1/2 mile from the bottleneck source compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



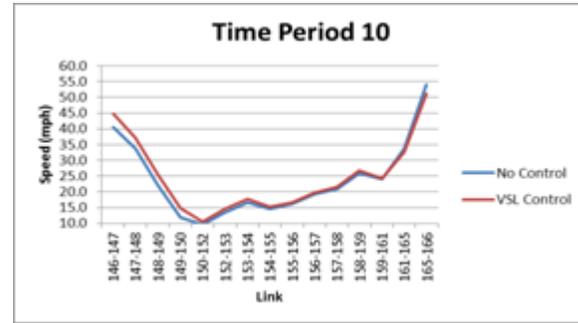
G



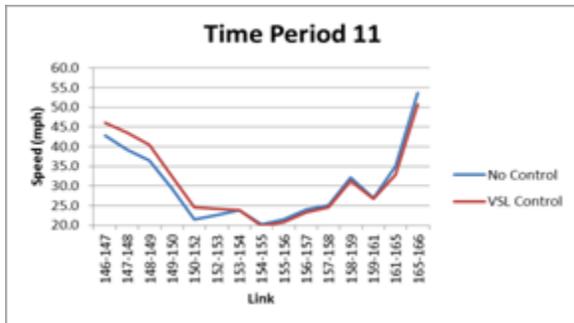
H



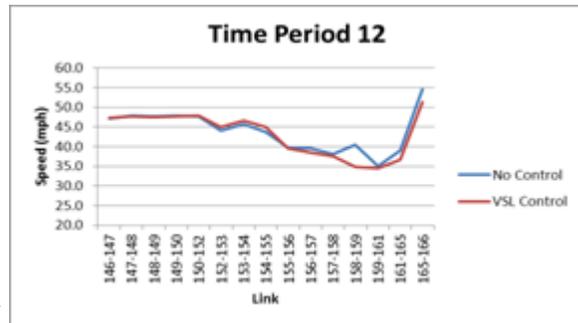
I



J

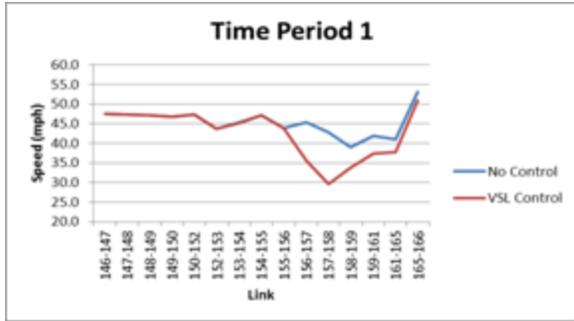


K

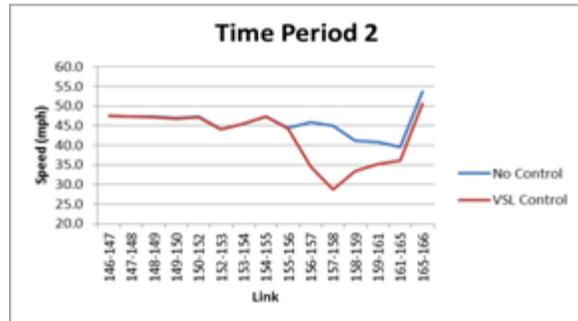


L

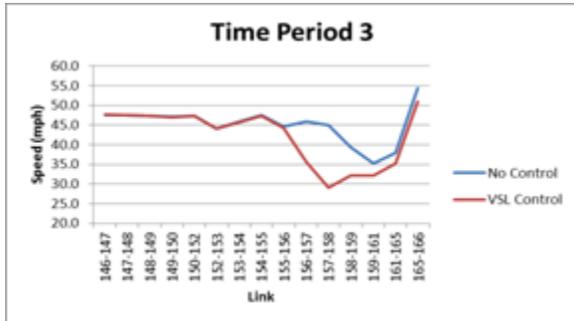
Figure C-21. Continued



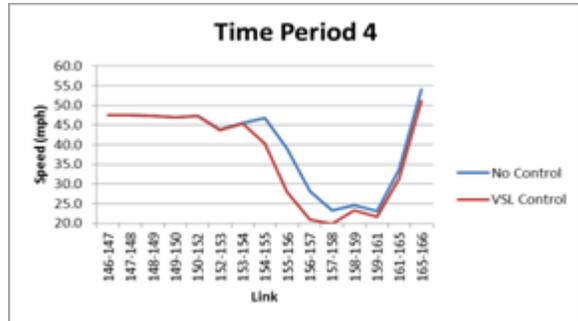
A



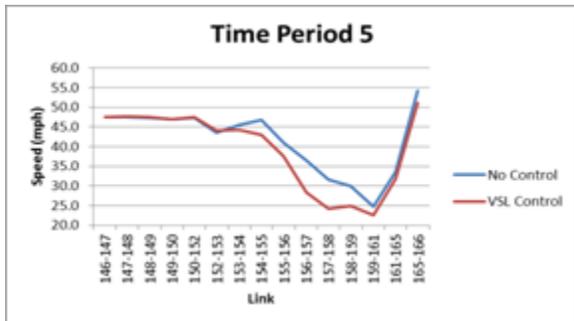
B



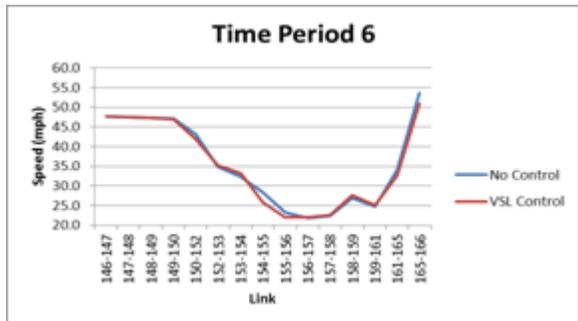
C



D

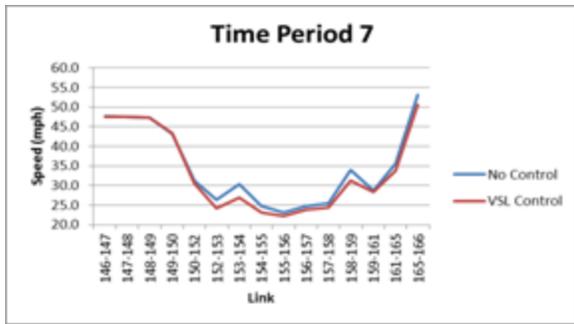


E

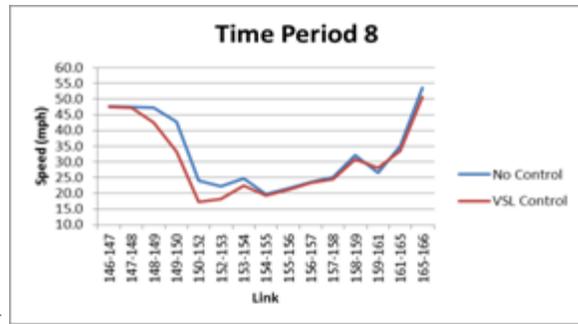


F

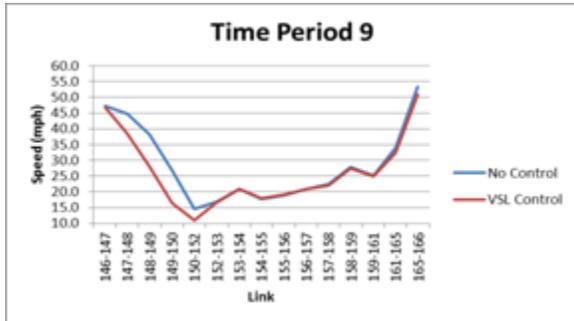
Figure C-22. Speed profile for multiple parameter--based algorithm using 1 sign spaced 1 mile from the bottleneck source compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



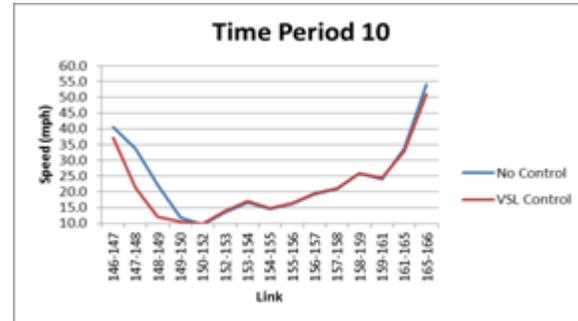
G



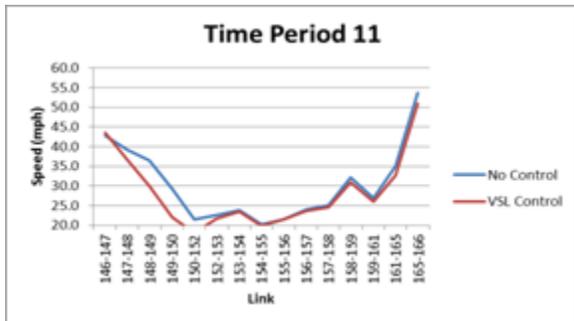
H



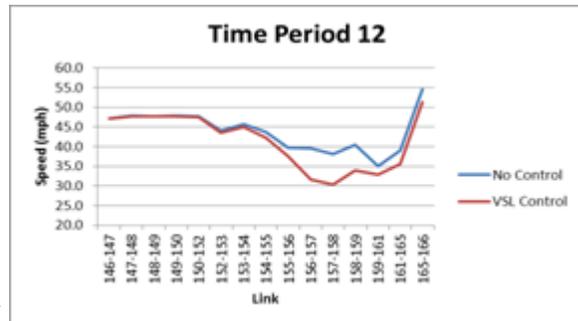
I



J

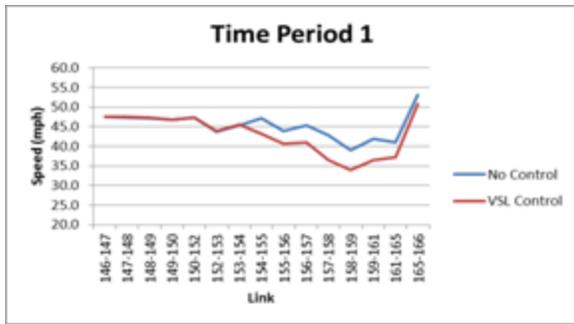


K

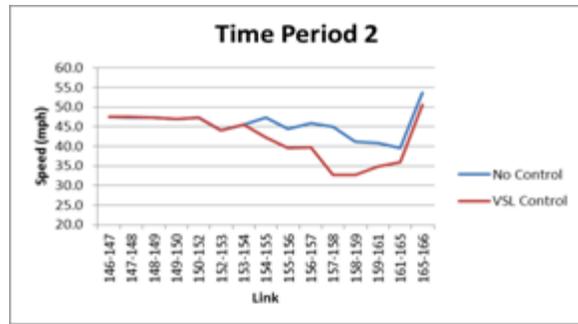


L

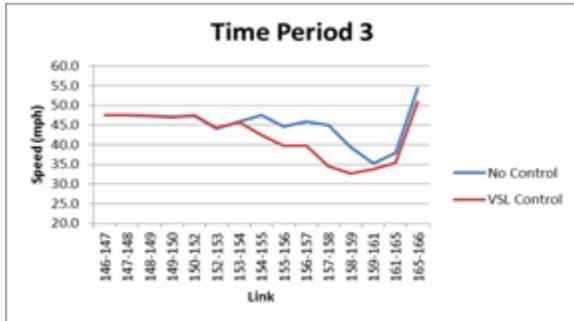
Figure C-22. Continued



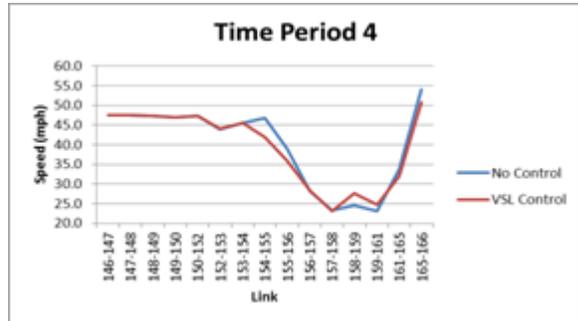
A



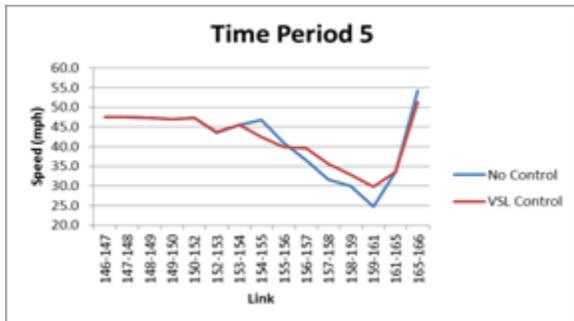
B



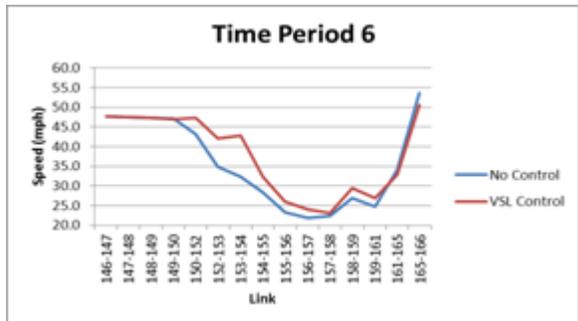
C



D

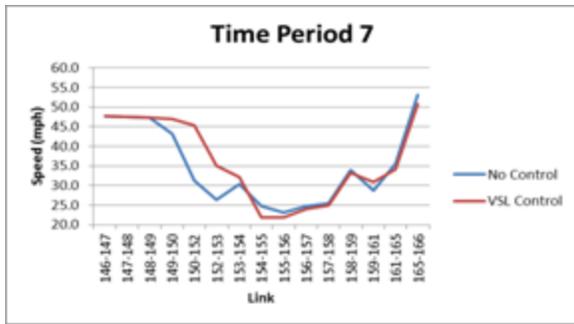


E

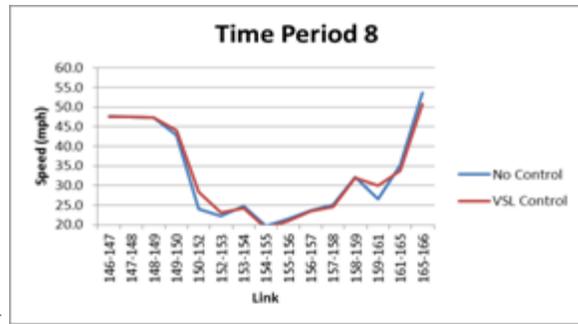


F

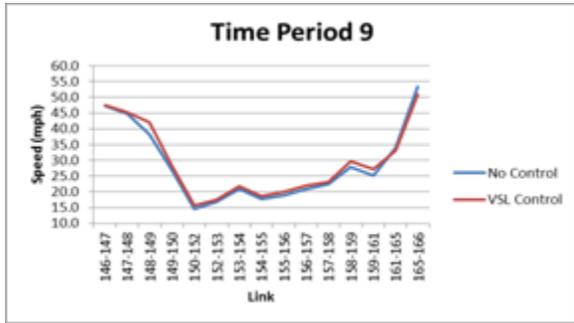
Figure C-23. Speed profile for multiple parameter--based algorithm using 2 signs spaced 1/2 mile apart compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time Period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



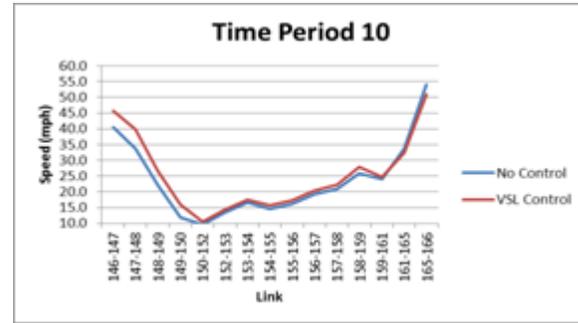
G



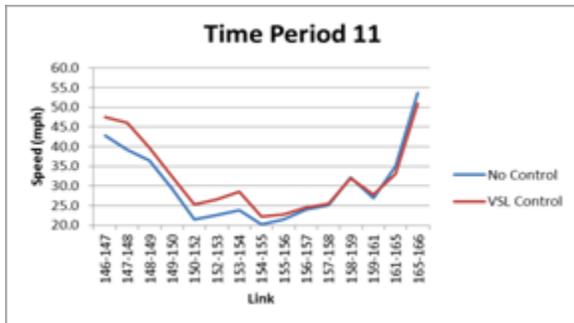
H



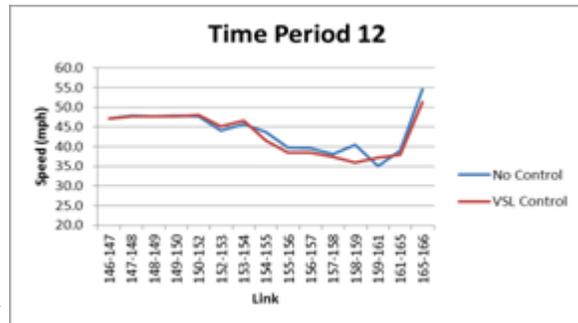
I



J

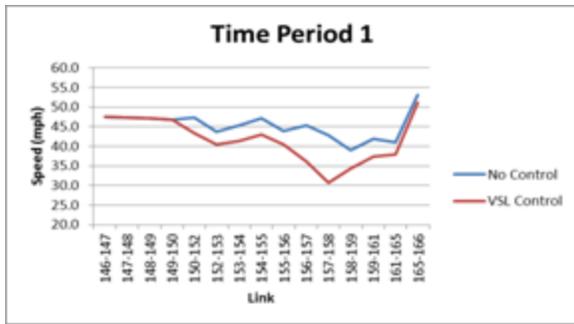


K

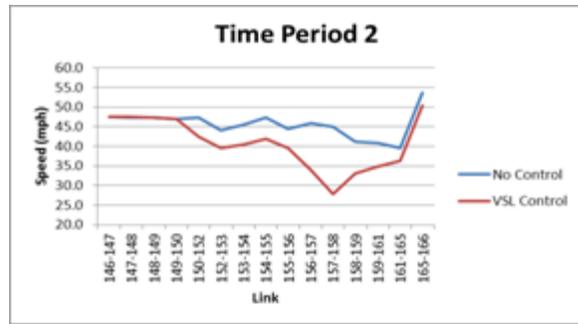


L

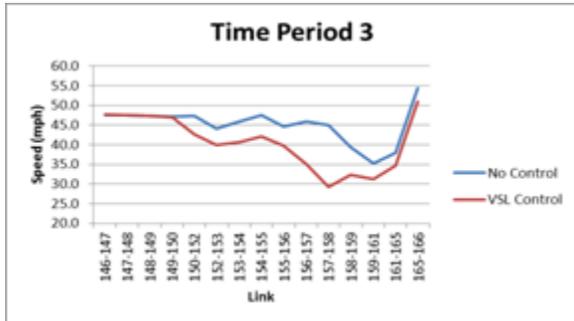
Figure C-23. Continued



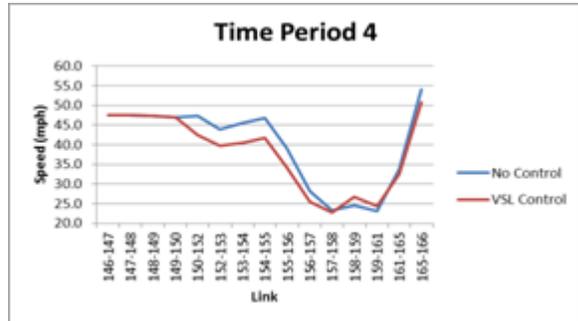
A



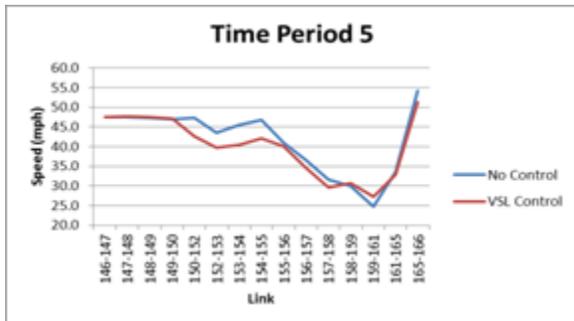
B



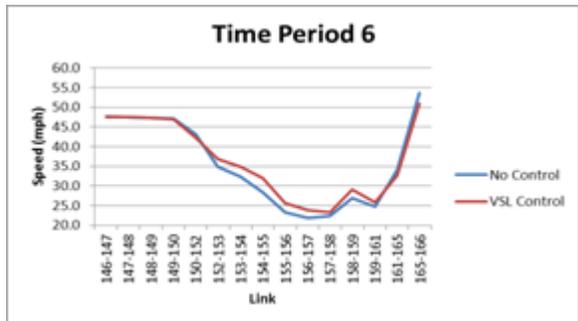
C



D

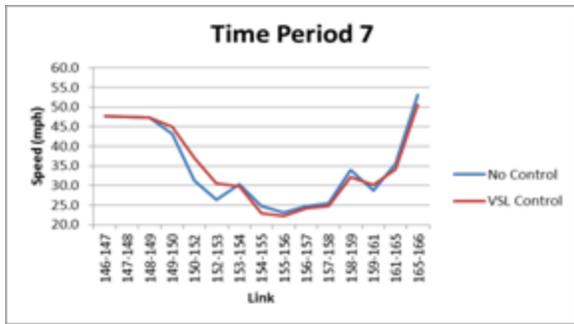


E

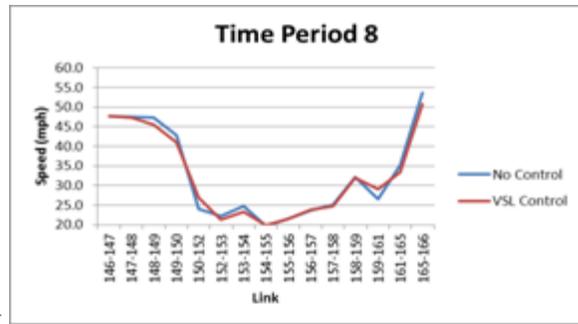


F

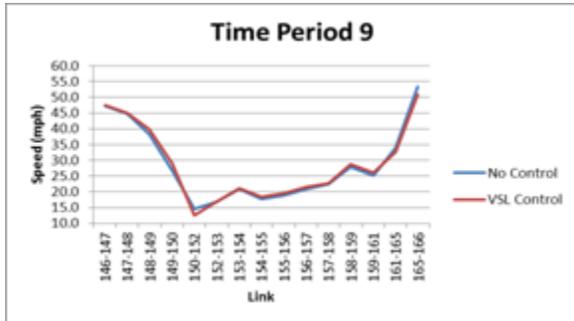
Figure C-24. Speed profile for multiple parameter--based algorithm using 2 signs spaced 1 mile from the bottleneck source compared to the no-control case. A) Time period 1. B) Time period 2. C) Time period 3. D) Time Period 4. E) Time period 5. F) Time period 6. G) Time period 7. H) Time period 8. I) Time period 9. J) Time period 10. K) Time period 11. L) Time period 12.



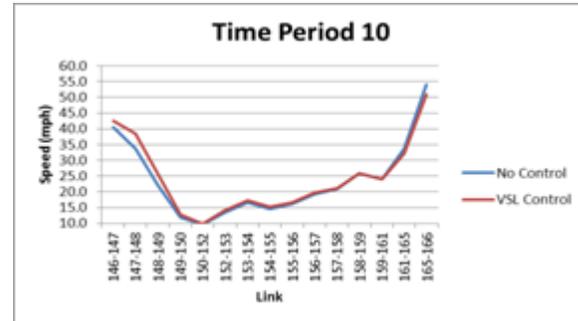
G



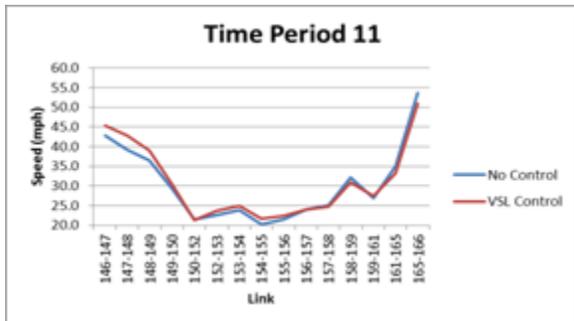
H



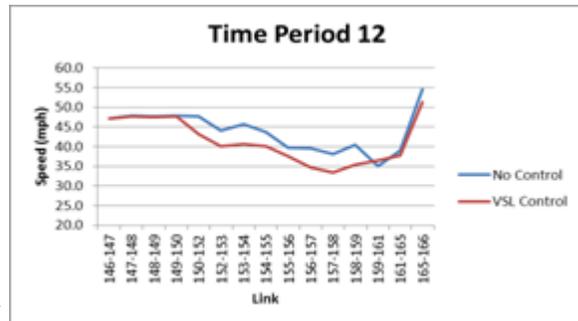
I



J



K



L

Figure C-24. Continued

LIST OF REFERENCES

- Abdel-Aty M., Dilmore, J., and Hsia, L., 2006. Applying Variable Speed Limits and the Potential for Crash Migration. *Transportation Research Record* 1953, pp. 21-30.
- Abdel-Aty, M., Dilmore, J., Dhindsa, A., 2006. Evaluation of Variable Speed Limits for Real-Time Freeway Safety Improvement. *Accident Analysis and Prevention* 38, pp. 335-345.
- Abdel-Aty, M., and Dhindsa, A., 2007. Coordinated Use of Variable Speed Limits and Ramp Metering for Improving Traffic Safety on Congested Freeways. In: 86th Annual Meeting of the Transportation Research Board, Washington, D.C., Paper No. 07-0008.
- Abdel-Aty M., Cunningham R. J., Gayah V. V., and Hsia L., 2008. Dynamic Variable Speed Limit Strategies for Real-Time Crash Risk Reduction on Freeways. *Transportation Research Record* 2078, pp. 108-116.
- Allaby P., Hellinga B., and Bullock M., 2007. Variable Speed Limits: Safety and Operational Impacts of a Candidate Control Strategy for Freeway Applications. *IEEE Transactions on Intelligent Transportation System* 8 (4), pp. 671-680.
- Boice, S., Bertini, R. L., Ahn, S., and Bogenberger, K., 2006. Dynamics of Variable Speed Limit System Surrounding Bottleneck on German Autobahn. *Transportation Research Record* 1978, pp. 149-159.
- Buddemeyer, J., Young, R.K., Dorsey-Spitz, B., 2010. Rural Variable Speed Limit System for Southeast Wyoming. In: 89th Annual Meeting of the Transportation Research Board, Washington D.C., Paper No. 10-2444.
- Carlson, R.C., Papamichail, I., Papageorgiou, M., Messmer, A., 2010. Variable Speed Limits as a Mainline Metering Device. In: 89th Annual Meeting of the Transportation Research Board, Washington D.C., Paper No. 10-1529.
- Florida Department of Transportation (FDOT), 2011. Managed Lane Operations – Adjusted Time of Day Pricing vs. Near-Real Time Dynamic Pricing. Tallahassee, FL.
- Elefteriadou, L., Brilon, W., Jacobson, L., 2011. Proactive Ramp Management Under the Threat of Freeway Flow Breakdown. NCHRP Report 3-87.
- Ghods, A.H., Rahimi-Kian, A., Tabibi, M., 2009. Adaptive Freeway Ramp Metering and Variable Speed Limit Control: A Genetic-Fuzzy Approach. *IEEE ITS Magazine* 1 (2), pp. 27-36.
- Goodwin, L.C., 2003. Best Practices for Road Weather Management, Version 2.0 Miretek Systems, Inc., Falls Church, VA.
- Haas R., Carter M., Perry E., Trombly J., Bedsole E., Margiotta R., 2009. iFlorida Model Deployment Final Evaluation Report, Washington, D.C. FHWA-HOP-08-050.

- Hegy, A., De Schutter, B., and Hellendoorn, J., 2003. MPC-based Optimal Coordination of Variable Speed Limits to Suppress Shock Waves in Freeway Traffic. In: Proceedings of the 2003 American Control Conference, Denver, Colorado. pp. 4083–4088.
- Hegy, A., De Schutter, B., and Hellendoorn, J., 2005. Model predictive control for optimal coordination of ramp metering and variable speed limits. *Transportation Research Part C*, 13 (3), 185–209.
- Highway Capacity Manual (HCM), 2000. Transportation Research Board. Washington, DC, ISBN: 0-309-06681-6.
- Hines, M., 2002. Judicial Enforcement of Variable Speed Limits. *NCHRP Legal Research Digest* 47.
- Jiang R., Wu Q., 2006. Suppressing Jams by Multiple Speed Limits: a Simulation Study Based on Cellular Automaton Model. In: Proceedings of the 84th Annual Meeting of the Transportation Research Board, Washington DC.
- Lee, C., Hellinga, B., Saccomanno, F., 2004. Assessing Safety Benefits of Variable Speed Limits. *Transportation Research Record* 1897 pp. 183-190.
- Lee, C., Hellinga, B., Saccomanno, F., 2006. Evaluation of Variable Speed Limits to Improve Traffic Safety. *Transportation Research Part C* 14 (200), pp. 213-228.
- Lin, P., Kang, K., and Chang, G., 2004. Exploring the Effectiveness of Variable Speed Limit Controls on Highway Work-Zone Operations. *Intelligent Transportation Systems* 8, pp.1-14.
- Lind, G., 2006. Weather and Traffic Controlled Variable Speed Limits in Sweden. *Movea trafikonsult*, Stockholm, Sweden.
- McLawhorn, N., 2003. Variable Speed Limit Signs for Winter Weather. Wisconsin Department of Transportation.
- The McTrans Center., 2009. Run-Time Extension User's Guide. University of Florida. Gainesville, Florida.
- Mitra A., and Pant P. D., 2005. A Framework to Evaluate the Impact of Variable Speed Limit Systems on Work Zone Traffic Operation Using VISSIM. In: Proceedings of the Institute of Transportation Engineers District 6 Annual Meeting, Kalispell, Montana.
- Papageorgiou, M., Kosmatopoulos, E., and Papamichail, I., 2008. Effects of Variable Speed Limits on Motorway Traffic Flow. *Transportation Research Record* 2047, pp. 37-48.
- PBS&J., 2009. I-4 Variable Speed Limit Effectiveness Study. Prepared for the Florida Department of Transportation, District 5.

- Piao J. and McDonald M., 2008. Safety Impacts of Variable Speed Limits – A Simulation Study. In: Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems, Beijing, China.
- Popov, A., Babuska, R., Hegyi, A., and Werner, H., 2008. Distributed Controller Design for Dynamic Speed Limit Control Against Shock Waves on Freeways. In: Proceedings of the 17th IFAC World Congress, Seoul, Korea, pp. 14060-14065.
- Placer J., 2001. Fuzzy Variable Speed Limit Device Modification and Testing—Phase II. Arizona Department of Transportation.
- Rämä, P., 1999. Effects of Weather-Controlled Variable Speed Limits and Warning Signs on Driver Behavior. Transportation Research Record 1689, pp. 53-59.
- Robinson M. D., 2002. Safety Applications of ITS in Rural Areas. U.S. Department of Transportation, Washington D.C.
- Robinson, M. D., 2000. Examples of Variable Speed Application. In: Proceedings of the 79th Annual Meeting of the Transportation Research Board, Washington, D.C.
- Sisiopiku V., 2001. Variable Speed Control: Technologies and Practices. In: Proceedings of the 80th Annual Meeting of the Transportation Research Board, Washington, D.C.
- Steel, P., R.V. McGregor, A.A. Guebert and T.M. McGuire., 2005. Application of Variable Speed Limits along the Trans Canada Highway in Banff National Park. In: Proceedings of the Annual Conference of the Transportation Association of Canada, Calgary, Alberta.
- Ulfarsson, G., Shankar, V., Vu, P., Mannering, F., Boyle, L., Morse, M., 2001. TravelAid, Washington State Transportation Center, University of Washington, Washington.
- Van den Hoogen, E., and Smulders, S., 1994. Control by Variable Speed Signs: Results of the Dutch Experiment. In: Seventh International Conference on Road Traffic Monitoring and Control. London, UK, pp.145–149.
- Washington State Department of Transportation. I-90 Two-Way Transit and HOV Operations – Variable Speed Limit Signs. [online] Available at: <http://www.wsdot.wa.gov/Projects/I90/TwoWayTransit/vsl.htm> [Accessed September 13, 2010].
- Zarean, M., Pisano, P., Dirnberger, K., and Robinson, M., 1999. Variable Speed Limit Systems: The-State-Of-The-Practice. In: Proceedings of the 1999 Rural Advanced Technology & Transportation Systems Conference, Flagstaff, AZ.

BIOGRAPHICAL SKETCH

Clark Letter was born in Melbourne, Florida in 1985. In 2009 he received his Bachelor of Science in civil engineering from the University of Florida, Gainesville, Florida. In 2011, he earned his Master of Science in civil engineering from the University of Florida. During his graduate studies, he was a research assistant for his advisor, Dr. Lily Elefteriadou.