5-2009

The development and evaluation of a detection concept to extend the red clearance by predicting a red light running event

Jun Xu
University of Tennessee

To the Graduate Council:

I am submitting herewith a thesis written by Jun Xu entitled "The development and evaluation of a detection concept to extend the red clearance by predicting a red light running event." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Civil Engineering.

Tom Urbanik, Major Professor

We have read this thesis and recommend its acceptance:

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Jun Xu entitled "The Development and Evaluation of a Detection Concept to Extend the Red Clearance by Predicting a Red Light Running Event." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Civil Engineering

Tom Urbanik, Major Professor

We have read this thesis
And recommend its acceptance:

Lee D Han

Chris R Cherry

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and
Dean of the Graduate School

The Development and Evaluation of a Detection Concept to Extend the Red Clearance by Predicting a Red Light Running Event

A Thesis Presented for
The Master of Science
Degree
The University of Tennessee, Knoxville

Jun Xu

May 2009

ACKNOWLEDGEMENTS

There are whole bunch of people I wish to thank for their help and encourage during the period that I am in University of Tennessee, Knoxville. First of all, I would like to show my gratitude to my supervisor, Dr Urbanik, a rigorous, respectable and experienced professor. He didn't only help me with this thesis but also with my daily study. He always showed me his great patient to help dissolving my problems. If one method wouldn't work, he would help me with all kinds of support till the problem solved. And his great knowledge and experiments inspire me all the time. Without his help, I couldn't even finish this thesis; without his help, I couldn't even finish my study.

Second of all, I would like to thank Dr Han, the first professor that I met in UTK. Since I've been in Knoxville, Dr Han was always there when I needed help. And he gave me lots of invaluable suggestions for my course studies, and even skills to learn English. I am definitely benefited from those suggestions.

Then, the third professor I would like to appreciate is Dr Cherry. When I was desperate with my midterm score of one of his class and about to quite, Dr Cherry's encouragement helped me insisting on that class and finish it. And also appreciate for inviting me to his house for Thanksgiving Day. That was the best Thanksgiving Day I've ever had.

I also have to thank Xiaoli Sun, Airton Kohls, Cuibi Liu and all of my friends for their encouragements, help and support. Finally, I owe my greatest thanks to my wife and my parents. I would never accomplish my study without their understanding, encouragement, and support.

ABSTRACT

This study focuses on developing and evaluating a detection concept to extend the red clearance by predicting a RLR event. It will dynamically extend the red clearance several seconds when a RLR is predicted to happen otherwise zero. Therefore the time will be used more efficient. In order to evaluate the influence caused by alternative detector positions, a VISSIM network was built up, connecting Econolite ASC/3 controller and ATACID. Due to the lack of realistic data, all the data in this study are fictional, but close to real. Several parameters are artificially modified in order to gain larger RLR occurrence. The two of the most crucial changes are the changing of reaction to amber signal and decreasing yellow interval.

“In this study, a MATLAB program predicts the RLR violation based on the data received from VISSIM via COM interface, and makes decisions。 And then, ASC/3 controller would execute every command received from the MATLAB program. Within every cycle, the MATLAB program would output data into a texture file, including the red extension type and red extension length. The result has four types: red clearance is extended while there is RLR violation (RERV); red clearance is extended no RLR violation (RENRV); red clearance not extended RLR violation (RNERV); red clearance not extended no RLR violation (RNENRV). RENRV and RNERV are two types of error that should be controlled in a reasonable range, especially RNERV. There are five scenarios while the position of the prediction detector is separately 100, 125, 150, 175 and 200ft away from the stopping bar. Each scenario has 5 runs with different simulation seed. By comparing the percentile of those four types of red extension among five scenarios, the system is more likely to extend all red as the distance increases; system accuracy will increase first and then decrease; Because detector located at 150ft has the least RNERV value, 2.1%, and least summation of RNERV and RENRV, 6.6%, it can be tentatively concluded that 150 ft is the appropriate position to locate the red extension detector, while the speed limit is 60 MPH in this study.

TABLE OF CONTENTS

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURE PAGE

CHAPTER I
BACKGROUND

According to data from FHWA (*1*), there are 106,000 crashes a year in the US caused by red light running, which result in nearly 1,000 deaths and 89,000 injuries. Almost 45% of drivers and passengers are injured by red light running. In order to prevent crashes caused by red light running, there are two principal methods. The traditional way is to station a patrol vehicle near an intersection. But this is dangerous to police men themselves and not the best use of available police resources. Also, the cost of this method is large, which is a burden on local finance. The second method is a red light camera. The red light camera is connected to a sensor which is installed above, on or under the surface of the pavement. While the sensor detects a vehicle passes through the detection zone during red, the red light camera will get a signal from it to capture several pictures of the license plate on the violation vehicle. Red light enforcement zones with red light cameras have been set up all over the world, including, the United States, the United Kingdom, the Netherlands, Israel, Austria and other countries. Red light cameras apparently have effects. For instance, In Oxnard, California, front-into-side crashes at intersections with traffic signals (the type of collision most commonly associated with red light running) were reduced by 32 percent. There were 68 percent fewer front-into-side crashes involving injuries (*2*). In Fairfax, Virginia, after one year of camera enforcement, violations were reduced by 41 percent (*3*).

Red light cameras affect some red light running behaviors, but not all of them. Red light running behaviors can be divided into two types: First, red light running which is executed by drivers who intentionally run the red light; and, second, red light running due to inattention or distraction. For instance, a vehicle driver runs red because he is distracted from driving, such as operating the A/C controls or changing a CD while the traffic light turns red. Also type 1 dilemma zone (short yellow interval) is another problem, which will also cause affect red light running. Obviously, a red light camera cannot improve unintential red light running like those two situations mentioned above. In the unintential case, some other approach is necessary to mitigate potential accidents.

In March 2008, Econolite released a new version of their ASC/3 software, version 2.40.00, which includes a new feature called the red extension option (MM-6-2). Siemens Energy & Automation Inc (Eagle EPAC) was doing something similar to it about the same time (*4*). Red extension extends the red clearance while a red light running behavior is detected, in order to avoid collisions and to protect drivers. This research is intended to develop a method to utilize the new feature to potential reduce red light running accidents.

STUDY OBJECTIVE

This research has two objectives:

1) Develop a detection concept to extend the all red when a possible red light running event is predicted;
2) Evaluate the effect of alternative detector locations on system performance.

CHAPTER II
LITERATURE REVIEW

Effect of Yellow-Interval Timing on the Frequency of Red-Light Violations at Urban Intersections

James A. Bonneson and Karl H. Zimmerman have discussed the effect of yellow-interval timing on the frequency of red-light violations at urban intersections in one of their papers. They pointed out that there are 3 characteristics (*5*): driver decision type, driver intent, and entry time of the red-light-running driver of red-light violation that affect operations. Driver decision type includes two cases which are avoidable decision and unavoidable one. Also there are two kinds of driver intent. One is intentional, the other is unintentional. The authors also define two entry times[5]. One is that driver enters the intersection during the first few seconds of red which might cause a left-turn-opposed; the other is that driver enter the intersection late into the red because what right-angle crash may happen.

Potentially, a long yellow interval can lead to bad habit because the yellow remains lit up as they arrive the stop line and are more inclined not to stop next time. But the data analysis result is quite different. All of the data are collected in field, using engineering countermeasures. The probability of stop of using 5s yellow interval is higher than using 3s yellow interval and 4s. They also found that reportable crashes are reduced by 8% (based on a study of 40 intersections) by increasing change interval, i.e., yellow and all-red intervals combined.

By analysis of the entry-time distribution, it indicates that more than one-half of the red-light running occurred in the first 0.5s of red, average enter time is 0.7s after the end of yellow. And about 80% of the drivers entered the intersection within 1.0s after the end of yellow. 80% of red-light violation entered within 1.0s after all-red interval. The most flagrant enter time was 14s after all-red interval.

A small increase in yellow interval duration or a reduction in driver speed can reduce left-turn-opposed crashes, and likely to have a more modest effect on red-light-related crashes. Improving driver attention and signal visibility is likely to be more effective at reducing red-light-related crash frequency. In general, red-light violations frequency will be decreased by at least 50% by 1.0s increase of yellow duration (not more than 5.5s).

Implementation of real-time yellow interval adjustment based on deceleration rates and pavement friction factors at signalized intersections

Another group of people did some similar research on yellow interval time impact too. Young-Jun Moon, Kwansoo Lim, and Yukyung Park published a paper, “Implementation of real-time yellow interval adjustment based on deceleration rates and pavement friction factors at signalized intersections” (*6*)*.* The methodology they adopted is using appropriate yellow interval time depending on the typical deceleration value that ITE recommends and pavement friction factor. By comparing the ITE formula and AASHTO formula, it is easy to find out that deceleration in ITE formula equals to g times friction coefficient (f) in AASHTO formula. And with all different friction coefficients given by AASHTO on different pavement conditions, i.e., wet and dry, and various approaching speeds, different deceleration rates can be achieved. Comparing them to ITE recommend value, the deceleration rates under wet pavement condition are higher up to approach speeds of 70 Km/h; the rates are slightly lower while the approaching speeds are higher than 70 Km/h; the rates on dry pavements are much higher than the ITE recommend value. The yellow interval that is calculated by using ITE typical value might not be realistic. The author suggest to use the real time yellow interval time which is calculated in the traffic controller by using the pavement friction coefficients that detected by the sensors for traffic and roadway surface conditions which is installed under the pavement of the roadway. Because of real time yellow interval time, higher driver compliance towards stopping at the onset of yellow will be gained. Not only yellow interval time will be effected by different pavement friction coefficients, but also stopping prediction in this project. This project will not concern the friction influence to yellow interval, but to stopping behavior. It becomes harder for drivers to stop while the pavement friction is lower. Under this condition, the red extension trigger value should be lowered to adapt the pavement condition or weather. This should be included in future work.

Impact of the signal control strategy on red light running

Not only yellow-interval will affect red light running, but also signal control strategy. This paper of Sophie Midenet, named as “Impact of the signal control strategy on red light running” concerns the impact of different types of control strategies on red-light running (*7*). All of the data are collected from a multi-camera observation system that automatically detects red-light running occurrences at several individual intersections. Those intersections are running under two different control strategies. The first control strategy is time-plan based strategy with vehicle actuated ranges on each approach. The second one is a real-time adaptive strategy named as “Cronos”, which uses an algorithm that optimizes the entire set of signal according to queue lengths on approaches and spatial occupancy rates on internal sections, measured every second by video sensors. These two control strategies impact differently on the traffic distribution of the link at the onset of amber and have different effects of the exposure to red-running occurrences. The author[7] thought that there should be an in-decision zone at some distance of the stop line. Most of drivers will go through the intersection if ahead of that in-decision zone and stop if behind. And the decisions of drivers inside that zone are divided. Also this article concerns the contextual factors that might influence the late red-running. For instance, surrounding traffic conditions on the link, upstream past clearing conditions, downstream flow conditions, and presence on the opposing link all should be considered. Analyzing occurrences of switchovers for different signal line and to reveal specific contextual features is possible because of the data base of traffic scenes collected under two types of control strategies.

Classification Analysis of Driver’s Stop/Go Decision and Red-Light Running Violation

Noor Elmitiny, Xuedong Yan, Essam Radwan, Chris Russo and Dina Nashar 2007 hold a field study of Driver’s stop/go decision and red-light running violation classification. In this study, a field data was collected in a high-speed signalized intersection, named as Alafaya Trail & Corporate Blvd, where a video-based system with three cameras was used to record the drivers’ behavior related to the onset of yellow. Several kinds of data, like drivers’ stop/go decision, red-light running violation, lane position in the highway, position (leading or following) in the traffic flow, vehicle type, and vehicles’ yellow-onset speed and distance from the intersection, are collected in this study. All of the data are extracted from a total of 36 one-hour videos including 28 off-peak hours and 8 peak hours. And 1292 vehicles’ behavior was analyzed and recorded. According to data analyses (*8*), the mean speed of vehicles with go decisions (M=49.8 mph) is higher than that with stop decisions (M=47.8 mph); the mean speed for leading vehicles (M=49.4 mph) is higher than that for the following vehicles (M=48.1 mph); the mean speed of vehicles traveling at the left lane (M=49.0 mph) and middle lane (M=49.1 mph) is higher than that at the right lane (M=47.5 mph); the mean speed of light-truck vehicles (M=48.4 mph) and passenger car (M=49.0 mph) are higher than that of large-size vehicles (M=45.5 mph); and the mean speed of red-light runners is 49.5 mph.

The classification tree models (8*)* were applied to analyze how the probabilities of stop/go decision and red-light running are associated with the traffic parameters. 80.9% of drivers would cross the intersection when the yellow-onset distances are smaller than 287.5 ft; 92.2% of drivers would stop if the yellow-onset distances are larger than 372.5 ft; If yellow-onset distances are between 287.5 ft and 372.5 ft, the stop/go decisions are more relied to the speed, 73.8% of drivers would stop when the speed is lower than 50.55 mph, 63.3 % would cross the intersection when the speed is higher than 50.55 mph; and speeding drivers are more likely to cross the intersection when they are the following drivers (74.2%) in the traffic flow. Also from the RLR data, drivers are less likely to run red light when the vehicles are in the leading positions in traffic flows, only 9.8%; For the following vehicles, 9.7% of drivers would run red-lights when the yellow-onset distances are smaller than 267.5 ft, 8.7% when larger than 372.5 ft. If vehicles are located between 267.5 ft and 282.5 ft, 72.2% of drivers would run red lights when the speeds are lower than 46.9 mph, 23.9% when higher; Between 282.5 ft and 372.5 ft, 19.3% of drivers would run red lights when the speeds are lower than 46.9 mph, 56.2% when higher. The result shows that distance and speed as two continuous variables are significant covariates associated with the yellow-entry time. The average entry time for leading vehicles is 3.8 sec, while that is 4.0 sec for following vehicles; 4.0 sec for light-truck vehicles, 3.9 sec for passenger cars and 4.2 sec for large-size vehicles. This article and previous one help to know factors affect red running behavior, and generate a rough idea of the project.

Evaluation of Driver Stopping Behavior on High Speed Signalized Intersection Approaches

Several external factors that will cause red light running violation have already been talked about. Internal factor, such as driving behavior, is also mentioned by a few people. Ihab El-Shawarby, Ahmed Amer, and Hesham Rakha did some research about it in their paper "Evaluation of Driver Stopping Behavior on High Speed Signalized Intersection Approaches, 2007". According to their paper (*9*), driver stopping behavior at the yellow-onset on high-speed signalized intersection approaches are characterized by using controlled field data gathered from 60 test subjects using an in-vehicle Differential Global Positioning System (DGPS). A total of 745 data records are ranging from a minimum time to stop bar (TTS) of

1.34s to a maximum of 6.19 s. By using statistical analyses, the impact of several aspects, like TTS, grade, age, etc, are investigated. According to the study (9), TTS at the onset of yellow, driver gender, driver age and grade don't impact driver perception time. But driver reaction time is dependent on the driver age and TTS at the onset of the yellow. The mean reaction time of drivers in the 40 to 59 age group is 0.34s, significantly lower than that of less than 40 age group (0.41s) and 60+ age group (0.38s). But younger group is ready to stop in a short time because they typically apply more aggressive braking rates. While the Perception-reaction-time (PRT) increases as the TTS at the onset of yellow increases, driver PRT won't be influenced by driver age and gender.

Analysis of Stopping Behavior at Urban Signalized Intersections: An Empirical Study in Korea

Another similar research was hold in Korea at the same year. Wonchul Kim, Junyi Zhang, Akimasa Fujiwara, Chang Nam Ryu, and Moon Namgung talked about the stopping behavior at urban signalized intersections in Korea in their paper—"Analysis of Stopping Behavior at Urban Singnalized Intersections: An Empirical Study in Korea". They said Korea has been made to install the traffic signals closer to the stop line for the purpose of dilemma zone protection[10]. And this article studies on the influences that the distance of signals to stop lines will make to drivers' behavior. They use two observation methods: one is on-site observation; the other is videotaped observation. The on-site observation includes 3 intersections in Jeon-ju City in Korea. First one ,which has signals installed 11.8 m far from the stop line (before-intersection case), has 3326 vph peak volume, 190 s cycle length, 3 s yellow, 117 s red; Second one ,which has signals installed 31.0 m far from the stop line (over-intersection case), has 2791 vph peak volume, 170 s cycle length, 3 s yellow, 99 s red; The last one, which has signals installed 67.0 m far away from the stop line (over-intersection case), has 1778 vph peak volume, 170 s cycle length, 3 s yellow, 123 s red. According to the observation result (*10*), it indicates that drivers' compliance with traffic signals significantly decreases as the distance between the signal and stop line increases, in other words the numbers of vehicle that enter the intersection at the onset of yellow and red will increase. The videotaped observation has a total of 29 hours of footage while 593 cycles were observed and only 198 were valid cases (stop: 90; cross: 108). The observation reveal that RTS (remain time to stop line at onset of yellow) in the before-intersection case is 1.36 times than the corresponding time in the over-intersection case which means drivers can always stop even at higher approach speeds in the before-intersection case. A binary logic model is established to represent drivers' stopping/crossing behavior at intersections to more quantitatively examine factors that influence drivers' behavior. And the result shows that signal location clearly has a significant influence on drivers' behavior. For instance, drivers are more likely to cross the intersection if the signals are located farther from the stop line.

Prediction of RLR from Discrete Point Detection

When I almost reach the end of this thesis, another paper related to this project, named as "Prediction of RLR from Discrete Point Detection" was published in TRB 2009 CD-ROM. Liping Zhang, Kun Zhou, Wei-bin Zhang and James A. Misener built up a probabilistic model to predict red light running (RLR) for collision avoidance systems at arterial signalized intersections by analyzing field data collected from an intersection in San Francisco. All data were captured by using AutoScope® Video cameras. According to their algorithm (*11*), vehicle distance to intersection, speeds of the approaching vehicles and time into yellow when vehicle is detected are important contributory factors to the model. They were using two discrete point detectors to estimate speed of vehicles. Additionally, AutoScope® Video camera is not only used to collect field data, but also to record RLR. As vehicles are treated as accelerating or decelerating at constant acceleration within the interval two detectors, acceleration can be estimated basing on the speeds detected by two detectors. In other words, as the speed and acceleration can be predicted, time needed for the vehicle to approach intersection can be predicted. Based on those information, whether vehicle will go through or stop can be estimated. So that system can make decisions depending on predictions. Here, authors defined two types of error, the missing report error when a RLR is reported as non-RLR (false negative), and also the false alarm error when proceeding through the yellow or stopping before the stop bar is accepted as RLR (false positive). In order to obtain a trade-off between these two types of error, decision boundaries are selected from Neyman-Pearson criteria based on field data. Authors compared the result from simulation of what parameters are calculated from the first set of field data with another set of field data. After that, they came out a conclusion that the closer the sensors, the better the performance. This conclusion is a little bit different from mine. It makes sense since I assume vehicles run in constant speed while they treat vehicles accelerate or decelerate at constant acceleration. And acceleration should be considered in future work to see if different conclusion will be gained.

CHAPTER III
METHODOLOGY

OVERVIEW

Due to the lack of realistic data on red light running and the difficulty of implementing experiments in the field, the project was designed as a simulation based study, using PTV VISSIM. VISSIM (*12)* is a microscopic, behavior-based multi-purpose traffic simulation program. This research used MATLAB as an external logic controller to determine when to extend the red clearance on an "ASC/3 controller", which will be used with a Hardware–in-the-loop interface (HIL) to VISSIM. HIL was chosen because it was the only feasible alternative at the time. The ASC/3 HIL approach requires a physical device that can make the ASC/3 HIL controller communicate with the computer running VISSIM. The HIL device selected in this project was the Advanced Traffic Analysis Center's Controller Interface Device (ATACID) (*13*). As shown in figure 1, computer is connected to "ATACID" via Ethernet port, and ATACID is connected to "ASC/3 HIL controller" via SDLC port. Through the interface device—ATACID, computer and traffic controller can communicate to each other.

The logic, discussed later, *o*f red extension used in the experiment is programmed and controlled by "MATLAB", which receives and sends information and data to VISSIM by using the COM feature in VISSIM as Shown in Figure 2.

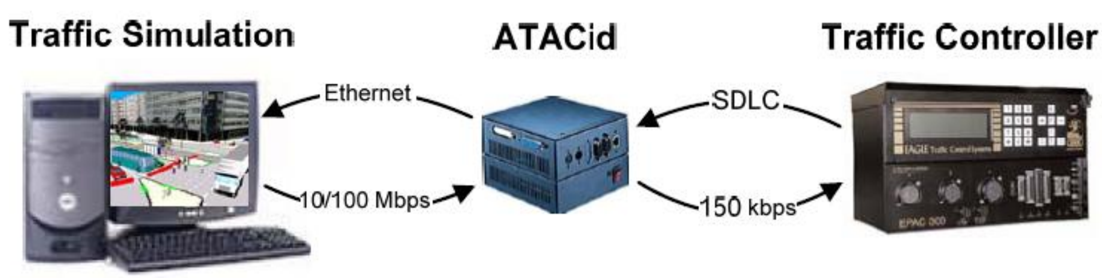


Figure 1 Connection between computer and traffic controller

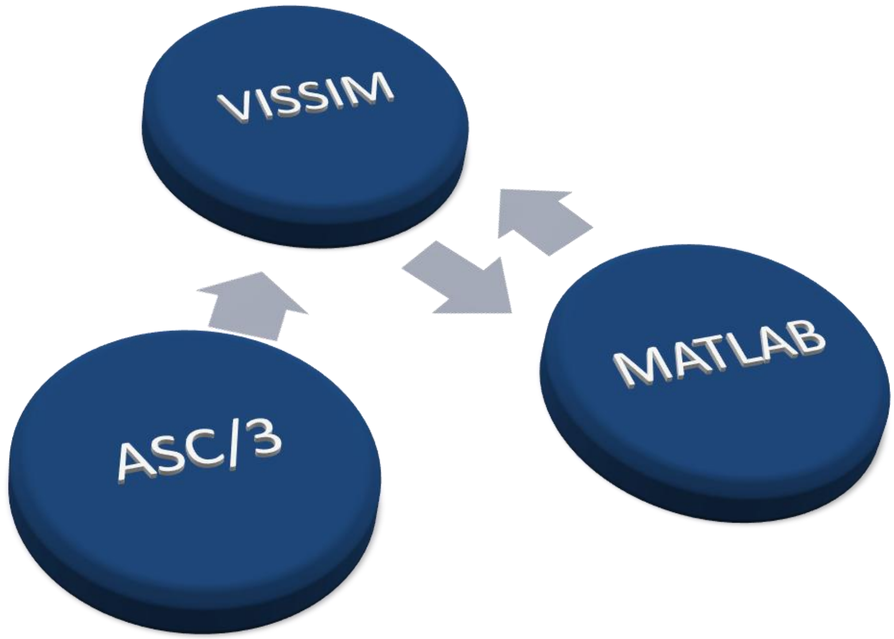


Figure 2 Relationships between ASC/3, VISSIM and MATLAB

It should be noted that there is another method that allows "MATLAB" to directly communicate with "ASC/3 controller" via Ethernet port, which can be done in the field, but not used in this experiment, but could be used to test in an actual field implementation.

ATACID

According to "ATACID" manual (*13*), the Advanced Traffic Analysis Center Controller Interface Device (ATACid) was developed to interface a NEMA TS 2 (2003) compliant traffic controller with a personal computer running a traffic simulation model to perform hardware-in-the-loop simulation (HILS). ATACID is simply connected to ASC/3 controller via SDLC, and to personal computer via Ethernet.

First, the ASC/3 controller needed to be properly configured. Under SDLC option (MM-1-4), as figure 3 shown, appropriate T&F and Detector BIUs (typically 1-4 for both devices) should be turned on, so that, ASC/3 controller can read and set virtual detectors in VISSIM. Then, the "Type 2 run as Type 1" option has to be enabled. Configuring the MMU shown in figure 4 is the last thing that is needed to set up the ASC/3 controller. After these procedures are implemented, the ASC/3 controller is successfully connected to ATACID.

Figure 3 T&F and BIU

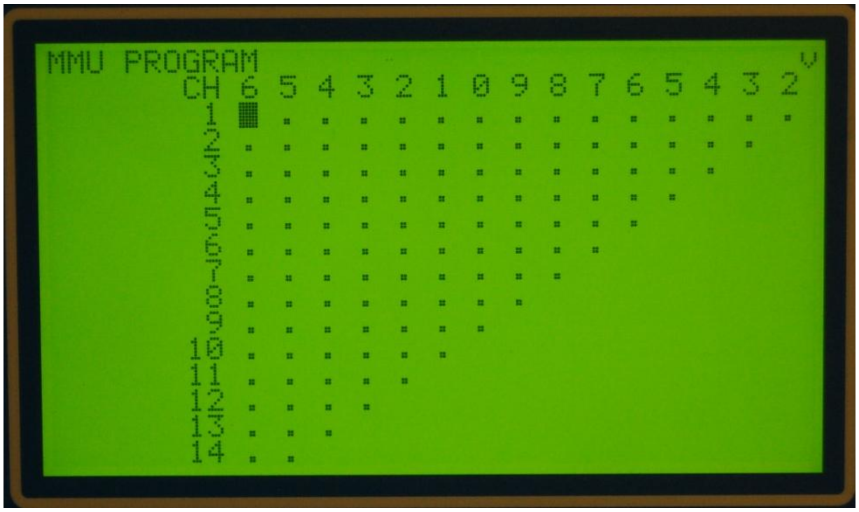


Figure 4 MMU program

The second step of setting up the ATACID is connecting ATACID and a computer via another type of cable, RS232 “Null modem” serial cable, needed for setup purposes only. The SerUpdt.exe program as shown in figure 5 is used for configuration. The first step is selecting proper COM port in order to make computer communicate with ATACID. In this project the ATACID is connected to computer directly via crossover cable, and manual mode is chosen instead of DHCP because of the direct connection. DHCP will be much easier if ATACID is within a network which has a DHCP server that can allocate correct IP address to ATACID. After choosing manual mode, pressing the retrieve button will show you the detailed information on the ATACID, such as IP address, Subnet Mask, Gateway and port number like shown in figure 5. Those items should be set as the same as computer’s, except for the IP address. However, both of the IP addresses have to be within the same section, which mean only last 3 digits can be different. Otherwise, they cannot be visible to each other. Lastly, the Port number has to be 2822, the default one. Configurations will be saved by pressing “Send configuration” button. Setup is almost done after procedures above are completed.

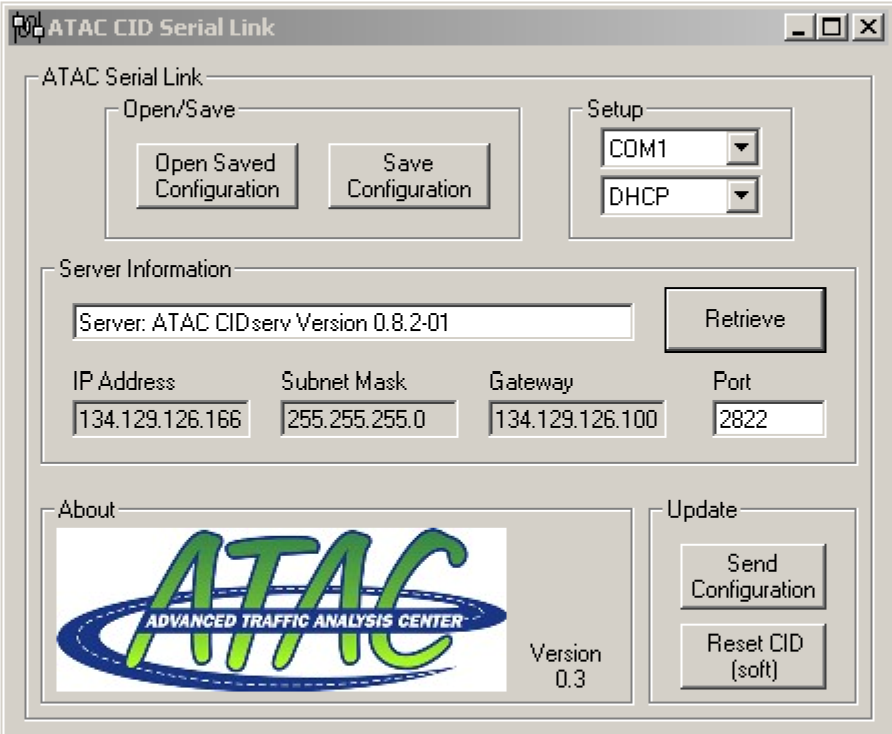


Figure 5 ATACID Serial Link

Then, next procedure is testing the connection of ATACID and computer. Locate and execute “CIDLink 1.1. jar” to test settings. “CIDLink 1.1. jar” requires a Java runtime environment to be installed on the computer. Under the “connection” menu of CIDLink, IP address and Port have been to exactly the same as they are set in previous step. The connection will be set up after pressing “run” under the “connection” menu. Then, if CIDLink looks similar to figure 6, that means the connection is successful. Testing won’t be done until the detector settings are checked. There are two ways to check the detectors: One is pressing the virtual detectors under the signal heads in CIDLink software; the other is running detector test scripts within CIDLink by pressing “detector test” under “Option” menu as shown in figure 7.

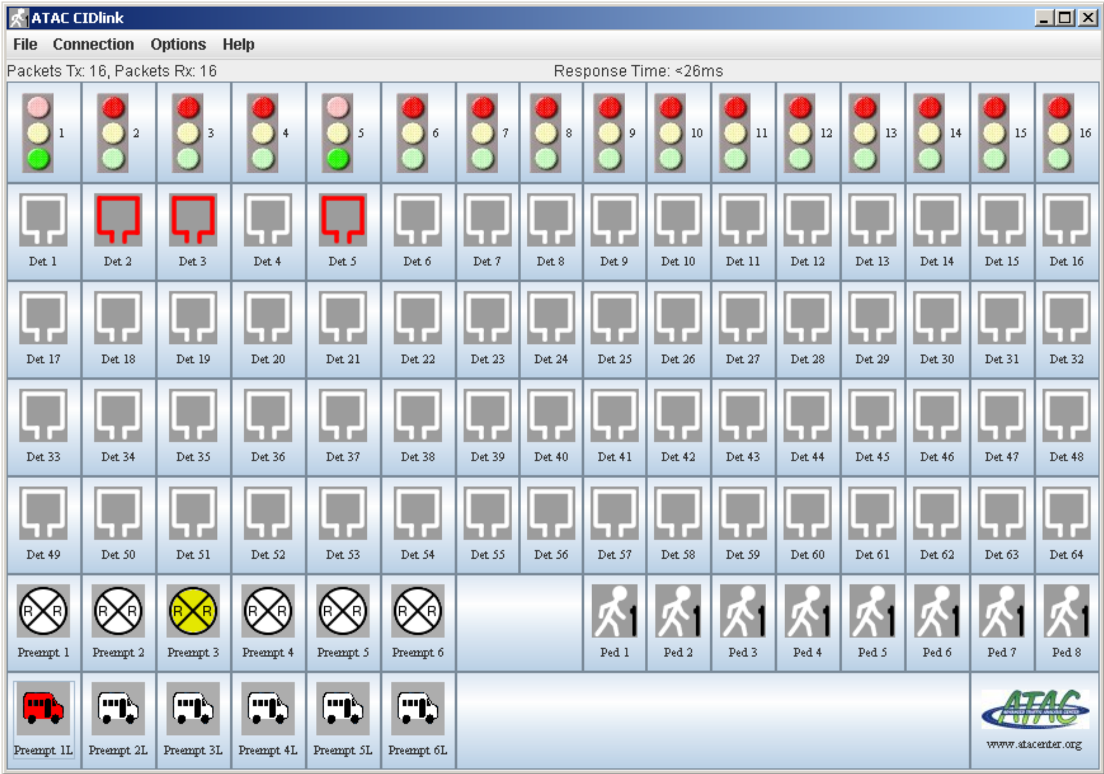


Figure 6 CIDLink interface

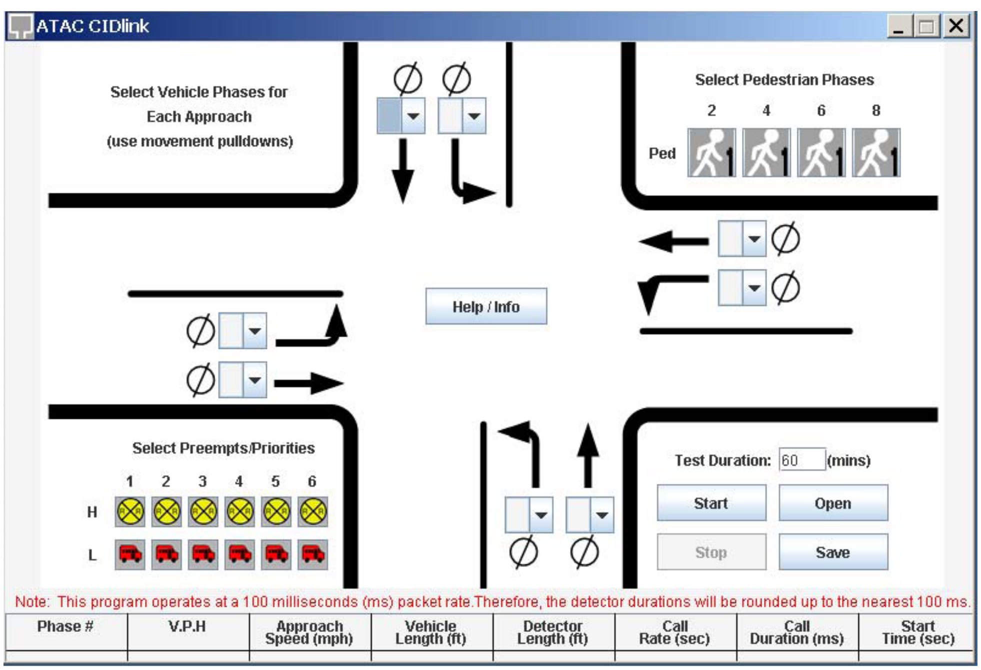


Figure 7 Detector test window

If the ASC/3 controller exactly reflects what CIDLink indicates while testing the detectors by any method mentioned above, computer and ASC/3 controller communicate well. Otherwise, return back to the first procedure to check whether ASC/3 controller is properly configured.

The last procedure is to copy several interface files to VISSIM directory and the working directory so that VISSIM can communicate to ATACID, and ASC/3 via ATACID. Two methods are given by ATACID to install those interface files. The setup.msi can install the DLL file to VISSIM directory automatically. Another way is to copy those four DLL file, MSVCP71D.dll, MSVCR71D.dll, SC_DLL1.3.dll, and SC_DLL1.3.wtt files into the VISSIM\exe directory that was created from the VISSIM installation. There are two more files, ATAC1.pua and TS2.vap, needed to be copied to the working directory, where the project locates. These files can allow VISSIM treat ATACID as a VAP controller. A few more settings have to be set before all components can work. Open ATAC1.pua in notepad and change the IP address and port to ATACID's.   Open VISSIM, change the signal control type under "Signal control" menu of the project to "VAP", as the figure 8 indicates, then save.

After all these procedures are completed, the setting and testing are finished.

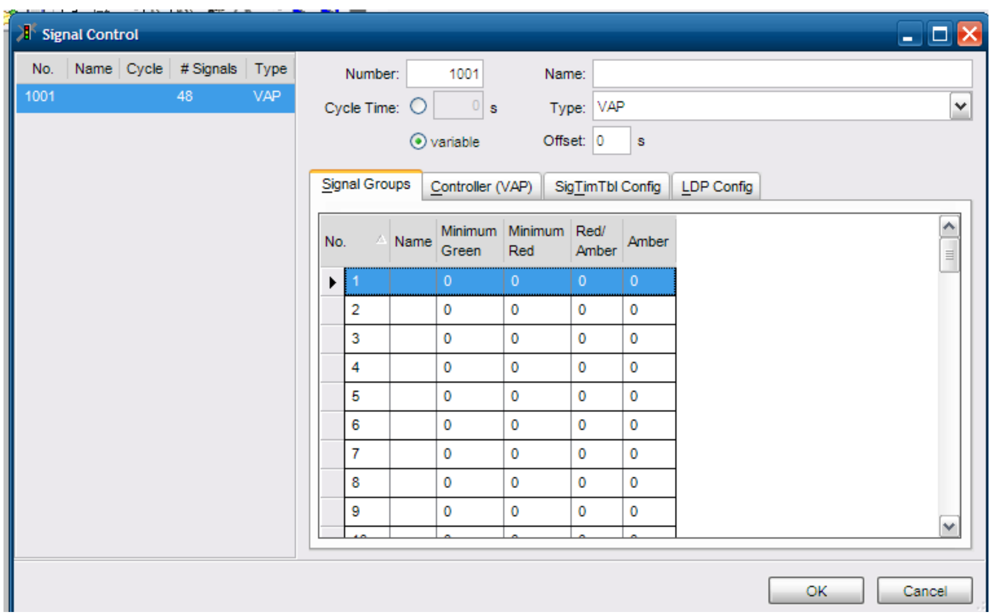


Figure 8 VISSIM signal control setting

VISSIM

3.3.1 Layout

The test intersection is shown in figure 9. This intersection includes 4 approaches begin as one lane that is divided into 3 lanes for one each for the left turn, right turn, and through. Because the purpose of this project is to find a red extension concept works and to compare the effects caused by different detector positions, this research project uses only one lane through movement on each approach with traffic volumes that can cause enough red light running events to analyze alternatives.

The width of the intersection is 114 feet. The length of each approach is long enough to avoid the congestion backward to vehicle generation point of the network, which could block the vehicles that are just generated in the network.

As shown in Figure 10, there are 24 detectors in the network. Each approach has a couple of detectors in front of stop bars. Detectors in the front are 7 feet away from stop bar, which are for calling phases, green extension. The other detectors are at variable distances from the stop bar and are used for red extension.

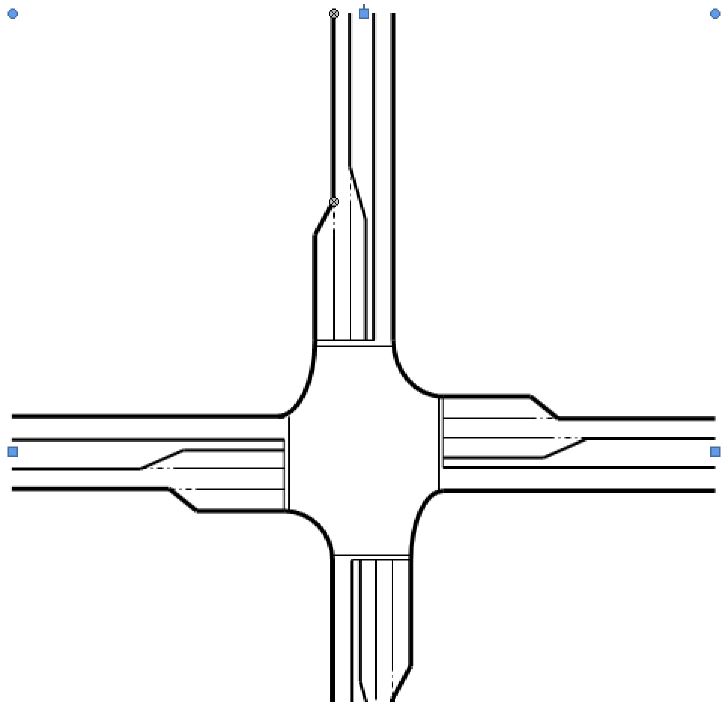


Figure 9 Network layout

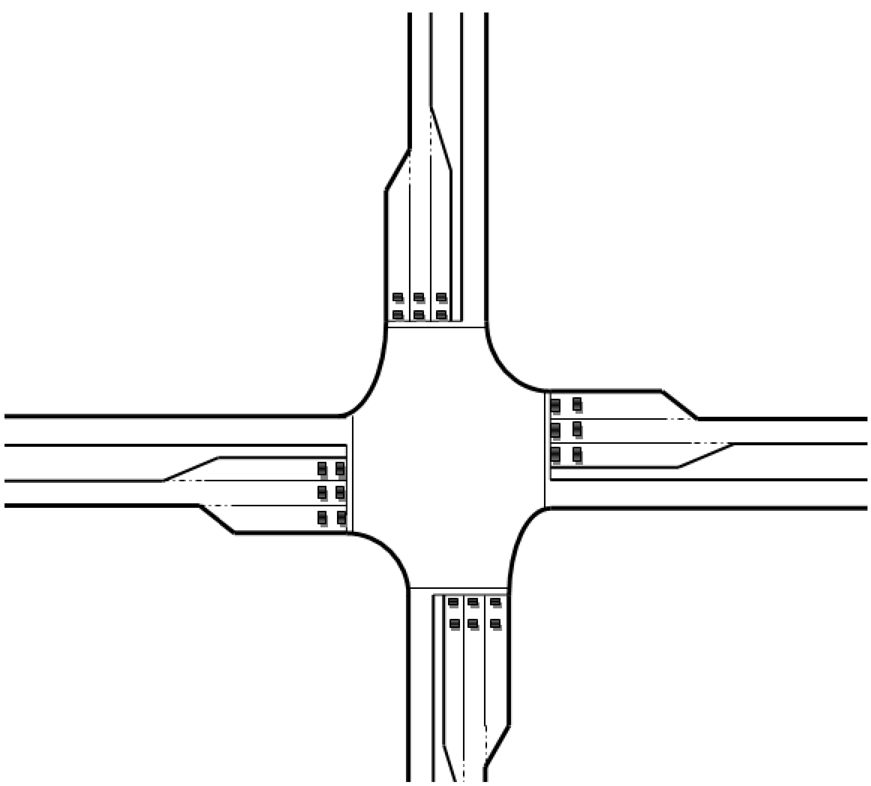


Figure 10 Detector positions

3.3.2 Network Parameter

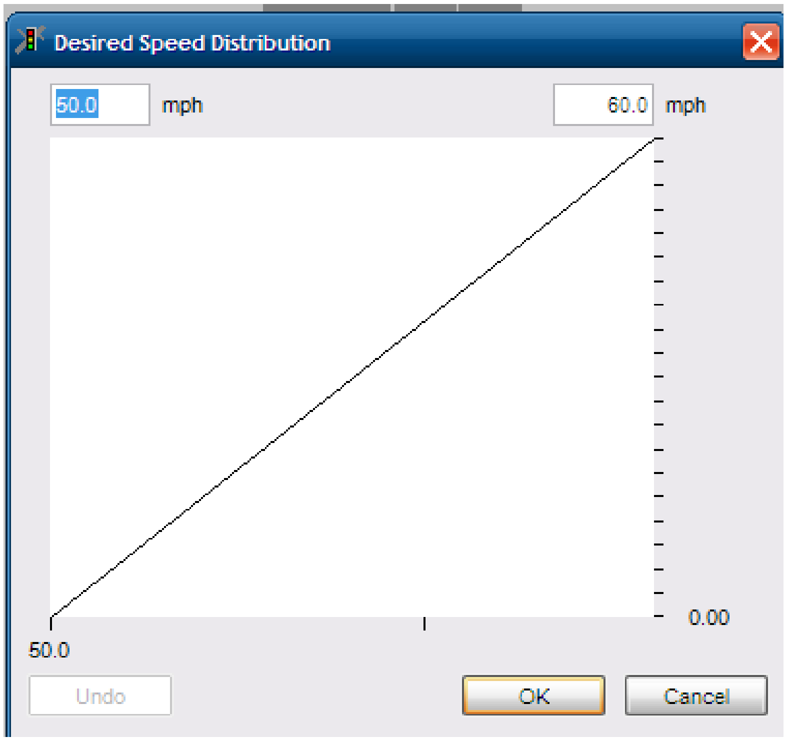


Figure 11 Desired Speed Distribution

The traffic in the network is composed only of cars in order to make the focus of the simulation on red light running and not more complex issues associated with trucks in the traffic stream. The desired speed distribution of vehicles was set at (50, 60 mph) in VISSIM, assuming the speed limit of this intersection is 60 mph. Two different speed values are required by VISSIM as shown in Figure 11 in order to have a small stochastic variation. The smaller value in the bracket is minimum value for the desired speed distribution, while the larger one is the maximum value. The main reason to use a higher speed and speed limit is to cause more red light running violations than would be seen at a lower speed.

The acceleration and deceleration distributions are shown in figure 12 and figure 13, which are the default setting of “VISSIM”. According to those two figures, there are three different curves showing three different values maximum, mean, and minimum. The maximum value of maximum acceleration and maximum deceleration is 11.5 ft/s^2 and -27.9 ft/s^2. And the minimum value of them is 6.4 ft/s^2

and -21.3 ft/s^2. The maximum value of desired acceleration and deceleration is 11.5 ft/s^2 and -9.8 ft/s^2 while the minimum value is 6.4 ft/s^2 and -8.4 ft/s^2. The desired deceleration value is consistent with the ITE value, 10 ft/s^2, which is used in the algorithm to predict when a vehicle is not going to stop.

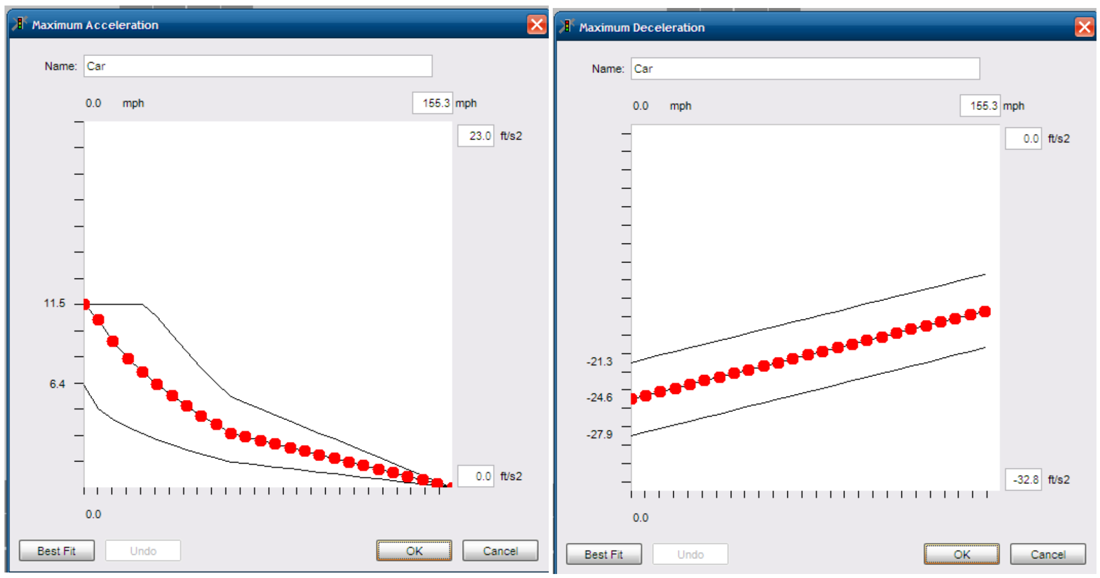


Figure 12 Maximum acceleration and deceleration

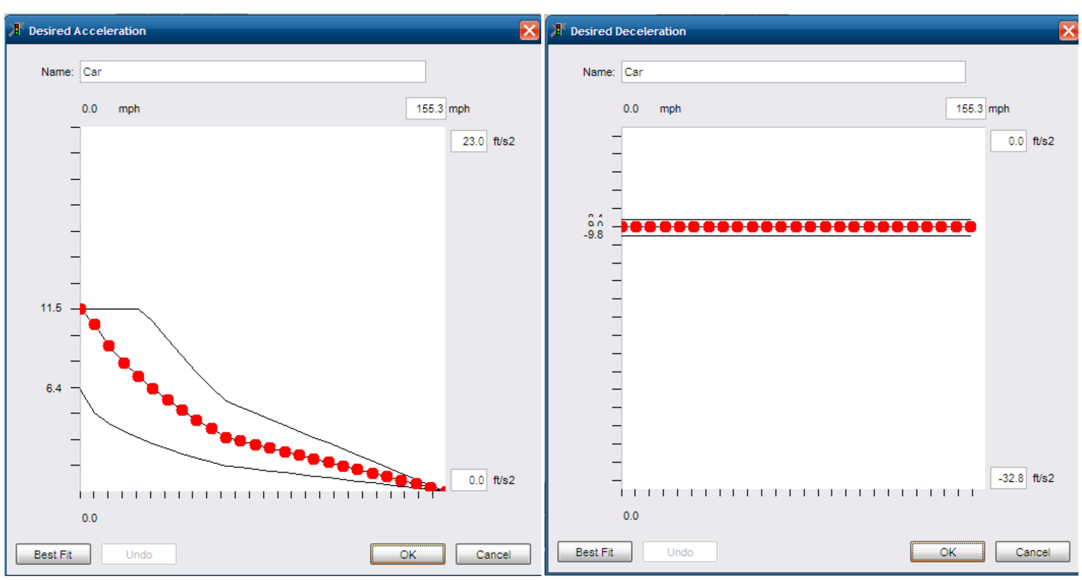


Figure 13 Desired acceleration and deceleration

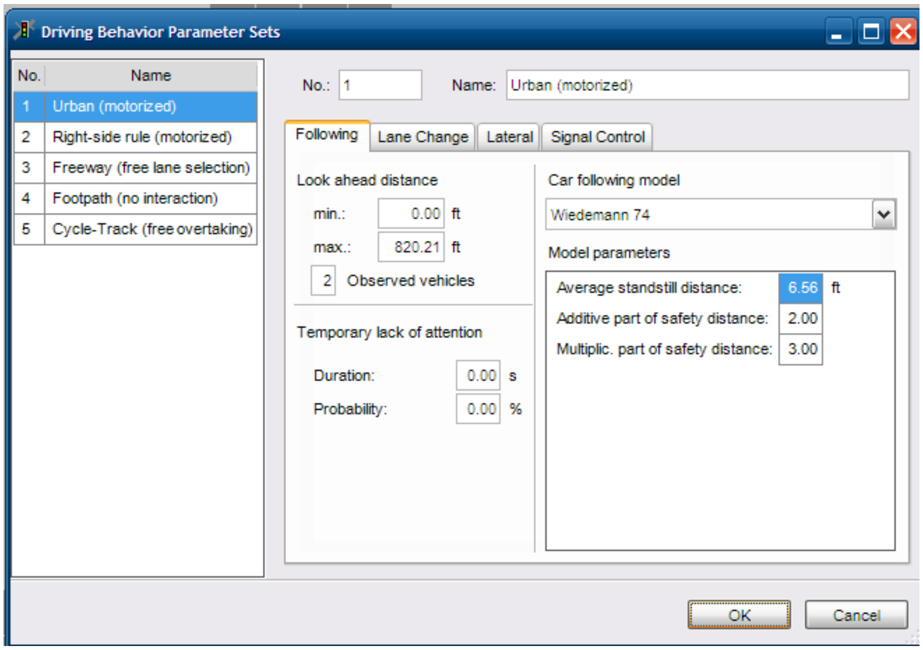


Figure 14 Driving behavior parameters

The driving behavior parameters are the default urban (motorized) settings in VISSIM. The driving behavior settings are not changed except signal control part.

3.3.3 The approach used to increase the number vehicles running the red

It was desirable to increase the number of red light running vehicles in order to have more occurrences. There are several ways to increase the number of vehicles running the red light, including increasing the speed, decreasing the deceleration, etc. But as the purpose of simulation is to detect and mitigate RLR, we selected two methods to increase violations. One was changing the driving behavior parameter, the other one is decreasing the yellow time. Figure 15 indicates a feature named as “reaction to amber signal” under “signal control” tab in “Driving behavior parameter sets”. According to the name, it could obviously affect the decision whether the vehicle should go through or stop. And the decision will only be made once and kept until the vehicle passed the stop line since it meets yellow light, after changing the decision model to “One decision”.

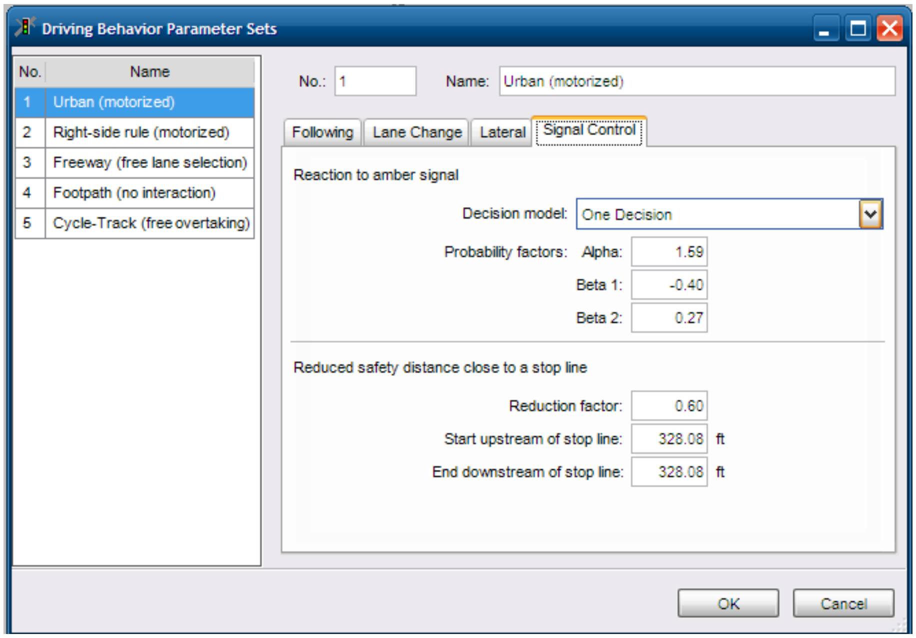


Figure 15 Reaction to amber signal parameter

The probability of this one decision at the amber light to stop can be calculated by equation 1. As mentioned above, the objective is to make vehicles run red. In other words, it will be successful if vehicles won't stop during yellow, even at the very last second of yellow. So lower probability of stopping at amber, more red running violations there will be. According to equation 1, there are three factors that will influence the probability, Alpha, Beta 1 and Beta 2.

$$p = \frac{1}{1 + e^{-\alpha - \beta_1 V - \beta_2 \, dx}}$$

Equation 1 Probability formula

For instance, assuming Alpha equals to 1.59, Beta 1 equals to -0.26, Beta 2 equals to 0.27, speed V equals to 60 mph and distance dx equals to 100 feet, the stopping probability is

$$p = \frac{\mathbf{1}}{\mathbf{1+e^{-1.59-(-0.26)*(60*1.47)-0.27*100}}} = 0.9965$$

Keeping other parameters the same, the stopping probability will decrease after decreaseing the value of Beta 1 to -0.40.

$$p = \frac{\mathbf{1}}{\mathbf{1+e^{-1.59-(-0.40)*(60*1.47)-0.27*100}}} = 0.0012$$

From the calculations above, it is obvious that stopping probability will be slight after decreasing factor Beta 1. Other factors, Alpha and Beta 2 do effect the stopping probability, but not as efficient as Beta 1. So Beta 1 is the only changed parameter in this project, which is -0.40. Others are as the same as the default value in VISSIM.

3.4.1 Red extension principle

The model of the ASC/3 controller being used in this project is the Econolite ASC/3-2100. It is running under software version V2.42.30, which has the latest red extension function, boot version V1.09.00 and configuration N3000. Software version V2.43.30 offers two methods to extend all red: One is to use the ECPI type of detectors, under MM-6-2. There are 4 types of detectors and type 3 is red extension detector. As shown in figure 16, after changing detector 52 to type 3, the red clearance will be extended as any vehicle is detected by detector 52. There are some issues related to type 3 detection that made the use of type 3 impractical for the research project. The “extend time” feature did not work when detector type is 3 according to results of several experiments that were run. So no matter how long the “extend time” is set, red extension phase will be terminated as the vehicle leaves detector 52. So for the purpose of extending all red phase dynamically, the use of the internal extend time could not be used. Therefore, another approach was developed.

The other approach is using the logic processor, under MM-1-8. A logic statement was implemented using MM-1-8-2, and then activated under MM-1-8-1. As indicated in figure 17, red extension on phase 4 will be on, if detector 52 is on, which means there is a vehicle on detector 52. Unfortunately, the logic processor has the operation as changing detector type, that is red extension phase will finish when the vehicle is no longer on detector.

Figure 16 Detector type

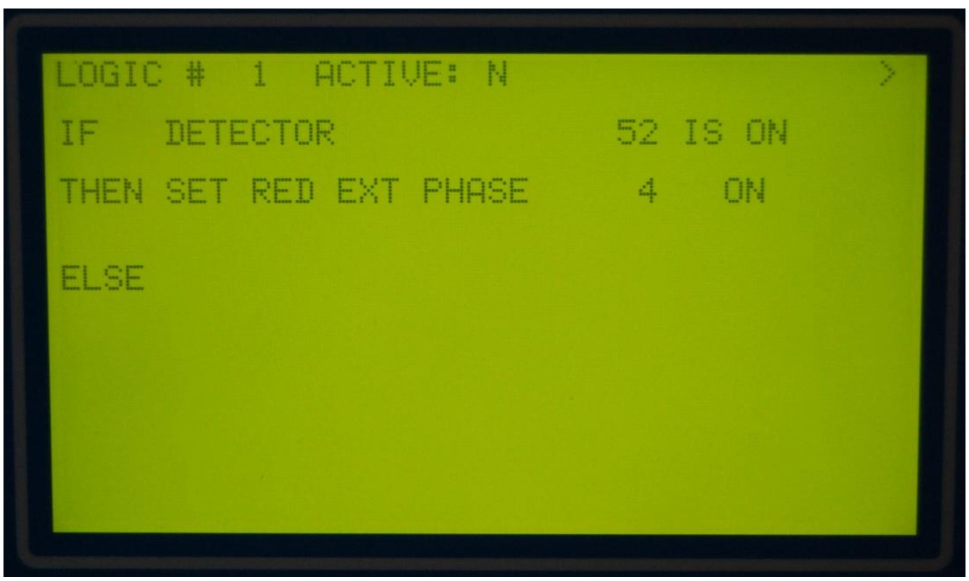


Figure 17 Logic statement

Since ASC/3 controller cannot provide a function to extend all red beyond the time the detector is occupied, other external software was needed to fulfill the dynamic red extension logic. In this project, “MATLAB” was adopted because of its capability and efficiency.

3.4.2 Time table

Table 1 Time Table

Phase	2	4	6	8
Min Green	5	10	5	10
Max Green	5	18	5	18
Passage Time	2	2	2	2
Yellow	3	3	3	3
All Red	0	0	0	0
Red Max	6	6	6	6

For simplicity, phase 4 is the only phase that was studied. Only four through phases were used, and all phases have no turning movements. In order to minimize time on the phases not under study, the other phases have relatively short minimum and maximum green times so that controller cycles back to phase ff4 faster as shown in table 1.

Larger minimum green times where used on phase 4 (and compatible phase 8) in order to allow vehicles on phase 4 and 8 to have a longer time to reach the desired speed, rather than terminating after enough time is provided to clear vehicles in the standing queue (which is the strategy on the cross street phases). The result of the timing selection is vehicles on phase 4 have more chances to meet yellow light.

The appropriate yellow time to prevent a type 1 dilemma zone is:

$$Yellow.Time = PIJR..Time + \frac{V}{44.1((\frac{a}{32.2}) \pm G)}$$

Equation 2 ITE yellow time calculation formula

Where
V: velocity of vehicle, usually use speed limits
a: deceleration
G: grade on the pavement

Using the ITE equation above, the yellow time should be 5.46 seconds when the speed limit is 60 mph on a ground level:

$$\boldsymbol{Yellow\ Time = 1 + \frac{60}{44.1 * \frac{9.82}{32.2} \pm 0} = 5.46\ seconds}$$

As we have already known the influence that short yellow interval causes higher incidence of red light running (*5,6*), a small yellow time value, 3 seconds, is used in this project. Therefore, vehicles are artificially put in the dilemma zone to artificially increase RLR occurrences.

3.4.3 Detector Settings

All detectors in VISSIM study network are 6'X6' loop detectors, except detector 51. All detectors are standard ECPI type 0 detectors, except detector 52. Detector 52 is the only type 3 detector used in the network for red extension. In order to avoid the influence from the vehicles in phase 4, detector 52 is off pavement as shown in figure 18 and only used by the external logic in MATLAB. Detector 44 gathers information, such as vehicle speed and time, and makes a decision, as described later, on whether to extend the red clearance or not using detector 52 to execute the decision. The position of detector 44 is changed to reflect the study's purpose of evaluating alternative positions. As figure 18 indicates, detector 51 is the only detector that is after the stop bar and only one which is not a 6' x 6' square loop. Its only purpose is counting during the red clearance for evaluation purposes. Therefore, the dimension of detector 51 is 0.1'X6' in order to count vehicles more accurately.

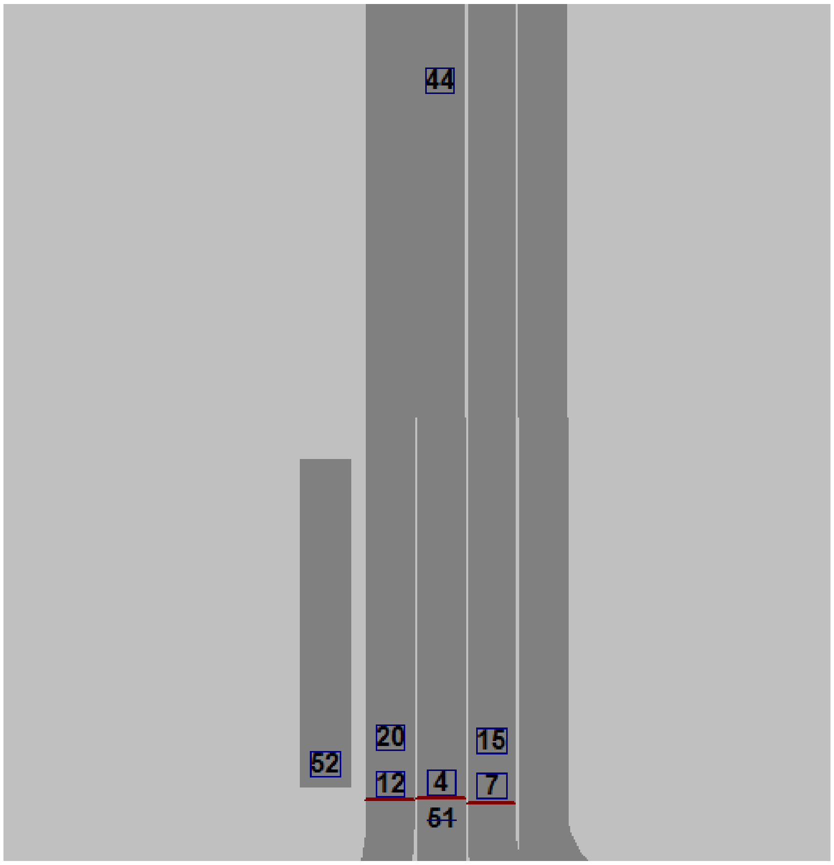


Figure 18 Detectors layout of phase 4

MATLAB

3.5.1 Algorithm

The section describes the development of the algorithm to extend the red clearance. This algorithm is constrained by how traffic signal controllers work. In order to extend the red clearance, we must know when it is going to start. Therefore, we cannot begin to implement the algorithm until the signal turns yellow. We then have the yellow time plus any fixed red clearance to make a decision to extend the red clearance.

The objective of this project is to extend all red while a violation of red light running is predicted. Assuming the decision is made instantly, in other words, it is made at the very moment that vehicle runs red, how long the decision window will be available depends on how long the all red is. For example, if all red time is 2 seconds, red extension can be only made while vehicles run red in this 2 second period. After that, red clearance is over and a conflicting approach turns green. Red extension is impossible begin after the fixed red clearance ends. Vehicles that run red at this moment are not protected. Because the goal of this project is to use a red extension phase instead of long red clearance when there are no vehicles approaching the stop.

Since a decision at the point a vehicle enters the intersection won’t bring success, prediction of RLR is necessary. However, the problem of when the decision should be made is still not resolved. Due to the flexibility of green time termination that can be between minimum green time and maximum green time, the exact ending time of green will never be known in advance, making it difficult to judge the probability that a vehicle violate red before the start of yellow. The yellow, interval, once begun, is suitable for predicting RLR as the time of yellow every cycle is fixed. The ending time of yellow can be easily figured out by knowing the beginning time of it.

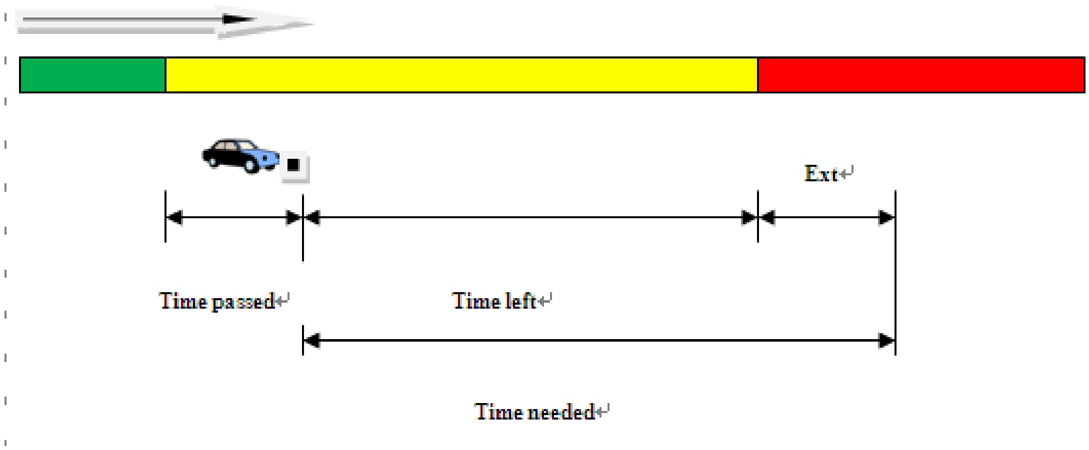


Figure 19 red extension theory

Figure 19 shows how the system makes a prediction. According to figure 19, the little dark square represents a loop detector upstream of the stop bar. Assuming in a cycle yellow interval of 3 seconds on phase 4 begins at time tb, , and the vehicle hit the detector at time t after start of yellow, then the time yellow that has already passed equals to the time that vehicle hit the detector minus the time that yellow interval started. For example, assuming tb equals to 14:30:42, and t equals to 14:30:44, the time passed tp equals 2 second.

tp=t-tb = (14*3600+30*60+44)-(14*3600+30*60+42)=2 second

Then the time yellow interval left tl equals to 1 seconds.

tl=3- tp=3-2=1 seconds

As soon as the vehicle hits the red extension detector, the detector begins to check its speed. The algorithm will compare it with the stopping speed that is calculated from ITE formula. For instance, assuming detector locates at 100 ft away from the stop bar, the computed stopping speed from this point equals to 30.25 MPH.

$$SSD = 1.47V * PIJR..Time + \frac{V^2}{30((\frac{a}{32.2}) \pm G)}$$

As PIJR time and grade are ignored in this study, the formula above can be transform into

$$V = \sqrt{\frac{30*a*SSD}{32.2}} = \sqrt{\frac{30*9.82*100}{32.2}} = 30.25 \text{ MPH}$$

If the detected vehicles speed is smaller than 30.25 MPH, which means this vehicle is able to stop, red extension time equals to zero. Otherwise, the vehicle will be assumed to keep on running at that speed as it assumed to be unable to stop. The difference between the yellow time left (Time left) and the time that the vehicle still need to reach the stop bar (Time needed) is the red extension time, assuming the fixed red clearance is sufficient to clear the intersection. It is assumed, if the vehicle won't stop, it will continue at the constant speed that red extension detector calculates. Therefore, time needed equals to the distance from the detector to the stop bar divided by vehicle speed. Time left equals to yellow time 3 second minus the time has passed since yellow begins to the vehicle hit the detector. Assuming the speed that detector detects is 45 MPH, which is larger than 30.25 MPH, time needed equals to 1.51 second.

tn=Distance/speed=100/(45*1.47)=1.51 second

If time needed is smaller than time left, which implies vehicle has enough time to pass over stop bar before the end of yellow, red extension time is not needed. Otherwise, red extension time equals to time needed minus time left. According to the example above, time needed equals to 1.51 second, which is larger than time left, 1 second, then extension time equals to 0.51 second.

Ext= tn - tl=1.51-1=0.51 second

Time passed: tp=t-tb;
Time left: tl=3- tp;
Time needed: tn=Distance/speed;
Red extension time: Ext= tn - tl
Where
tp: time has passed
t: current system time
tb: the time that yellow begins at
tn: time that is needed for vehicle to reach the stop bar from the current point

Ext: red extension time

Equation 3 Red extension time calculation

3.5.2 VISSIM COM Interface

MATLAB is the computational engine for this project, since it calculates information gathered by detectors and computes the dynamic red extension as an external logic processor. It gathers data directly from VISSIM and sends commands to VISSIM via the COM interface, which allows VISSIM to run with other applications. Thus, the COM interface allows an application to read status of objects in VISSIM and then send commands back.

The VISSIM COM object model is based on a strict object hierarchy. Some certain rules should be followed to access different lower-level objects. As figure 20 and 21 indicate, VISSIM is the highest object which other objects belong to, such as Net, Simulation, Evaluation, etc. For example, if link is the object that we want to achieve, then a command like this, “hVissim.Net.Links.Link” , should be input in Matlab.

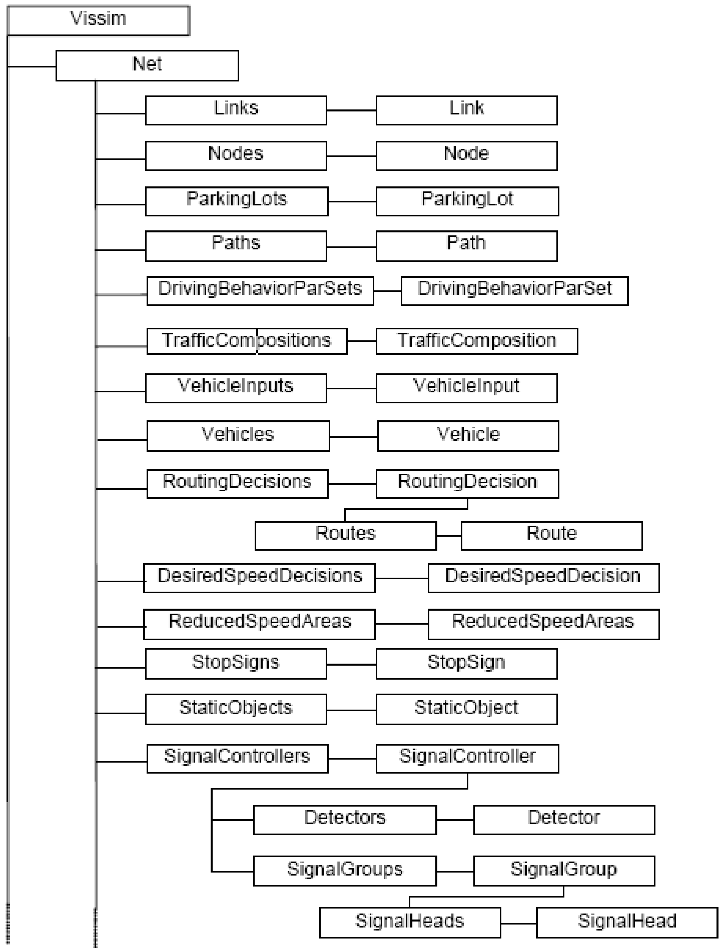


Figure 20 VISSIM Object Model

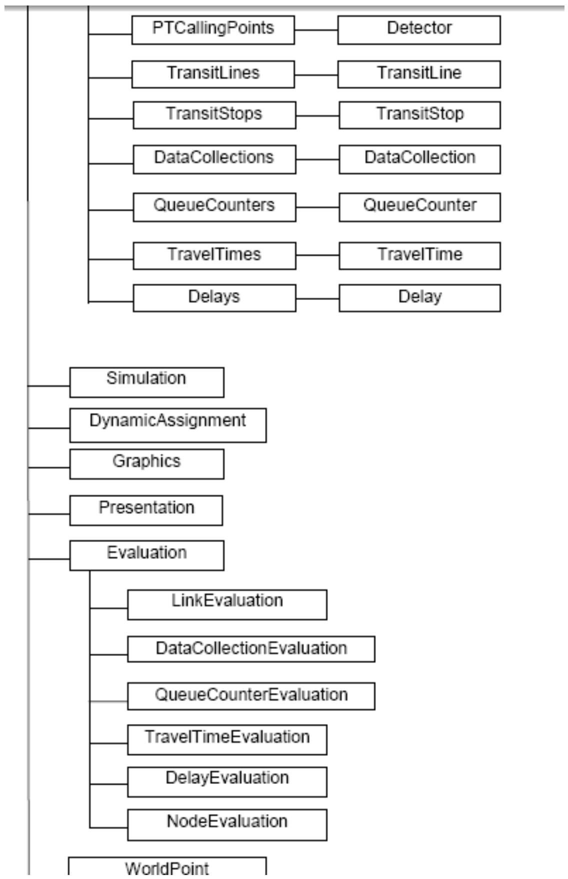


Figure 21 VISSIM Object Model continued

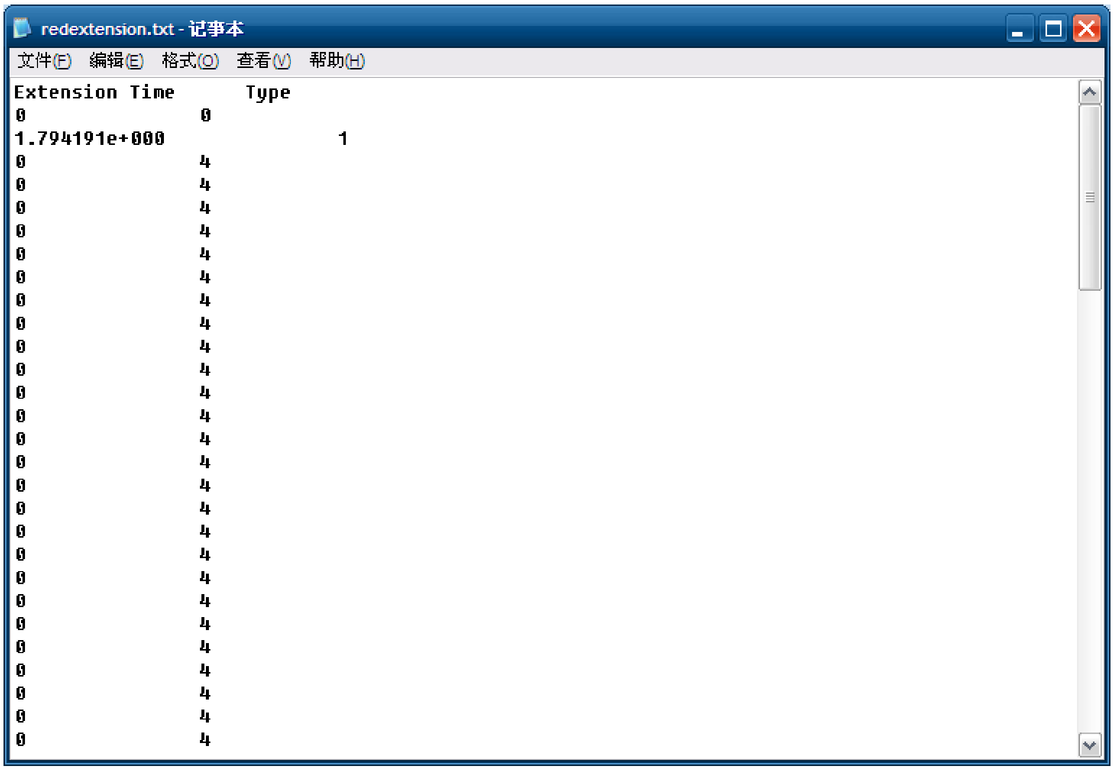
Implementing the algorithm mentioned above, only three objects that are used detector, signal head, and simulation, are the in this project. There are several properties that detector object provides in VISSIM, including, detection, headway, impulse, vehicle ID, vehicle speed, etc. By using the command “hVissim.Net.SignalControllers .GetSignalControllerByNumber (1001). Detectors .GetDetectorByNumber(51). AttValue('DETECTION')”, detection by detector 51 can be read by Matlab. The command above shows that object Detectors belongs to object SignalControllers, which belongs to object Net. “GetXXXByNumber()” is used to get information of any specified object, like detector or signal controller, by its number. And “AttValue(‘XX’)” is used to access the properties that an object provides. For example, “AttValue('DETECTION')” is to access the detection status of detector; AttValue('SPEED') is to access the speed of the vehicle that is on that detector.

There are two methods that Matlab can control the way VISSIM runs the simulation. One is continuous simulation; the other is single step simulation. Because Matlab has to gather data from VISSIM, make calculations and make predictions each cycle, the single step simulation method is be better for this application. A single step simulation of 0.1 second is used, so Matlab calculates every 0.1 second and runs the next single step using the command "hVissim.Simulation.RunSingleStep". There is a loop to run single step simulation again and again in order to keep simulation running unless Matlab jump out of the loop. The loop will only be exited out when simulation period reach the pre set value.

After the simulation starts to run, Matlab will check the signal status of phase 4 first to see whether it turns to yellow. As soon as phase 4 turns to yellow, Matlab will record the current time as yellow begin time-- tb. Then detector 44 will be checked. If there is a vehicle on detector 44, Matlab inquires VISSIM the speed of this vehicle, and record the current time as the time that vehicle hit the detector—t. By using the algorithm mention before, Matlab will give a value to variable "Ext": Ext will be zero If current speed is smaller than stopping speed, or time needed is smaller than time left though current speed is larger than stopping speed; and it will equal to the difference between time needed and time left, while vehicle doesn't have enough time left to pass over the stop bar by the end of yellow.

3.5.3 Output

Though the system now can make judgments, performance data is needed to check accuracy of judgments and efficiency of the system. As the VISSIM evaluation metrics don't match the requirement of this project, Matlab is used to generate a txt file which records data as shown in figure 22 below. Only the extension time and extension type are recorded. Here, extension type is divided into 4 types: Type 1, all red is extended while there are red light running vehicles, defined as RERV; Type 2, all red is extended while there is no red light running vehicles, defined as RENRV; Type 3, all red is not extended while there are red light running vehicles, defined as RNERV; Type 4, all red is not extended while there is no red light running vehicles, defined as RNENRV. Type 1 and type 4 results are what we desire, while type 2 and type 3 are errors that should be limited. The linear detector 51 will only detect red running vehicles when phase 4 is red. If extension time has a value and detector 51 has detections during red, then type 1 will be recorded; or if extension time has a value, but detector 51 has no detection, type 2 will be recorded. Similar, type 3 or type 4 will be recorded if appropriate situation happens. But these data are still recorded in memory, not in the output file yet. Data won’t be written to output file until conflicting phases, phase 2 or phase 6 turns to green. No matter how many vehicles run red during this period, they will be treated as one event each cycle.




```
Extension Time      Type
0               0
1.794191e+000            1
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
0               4
```

Figure 22 Output format

CHAPTER IV
EXERIMENTS, RESULTS AND DISCUSSION

EXPERIMENTS

As one of goals of this project is to figure out the effects of different locations of the red light extension detector (detector 44), 5 scenarios were selected for evaluation. These 5 points are respectively 100ft, 125ft, 150ft, 175ft, and 200ft away from the stop bar. Every scenario has 5 simulation runs with 5 different simulation seed, such as 10, 20, 30, 40, and 50. Each run lasts 1 hour. And those 5 simulation seeds used in 5 scenarios are totally the same. Though every scenario has exactly the same simulation seeds and other simulation parameters such as traffic volume, simulation speed, simulation period and etc, every scenario cannot have exact same occurrences of red extension due to the red extension influences. For instance, system doesn't extend all red in scenario 1, while red extension time is 1 second in scenario 2 at the same time point. Then, phase 2 and 6 turn green 1 second later in scenario 2 than in scenario 1 which means traffic conditions in these two scenarios become different after this every second. So number of cycles in each scenario will be a little bit different from each other, though they are running under exact same conditions except different position of detector 44. Number of output samples will be different in each scenario because data will only be written to the output txt file once in each cycle as the previous algorithm said.

RESULTS

Table 2 shows number of RLR occurrences. The results show that the numbers of occurrence for each of the scenarios are quiet close to each other, with only slight differences, even seed to seed. Since the sample sizes are slight different, comparing the occurrence frequencies of each red extension types between different scenarios cannot be precise. Four types of red extension, RERV, RNERV, RENRV, and RNENRV, are defined in this project as table 4.1.2 indicates.

Table 2 Times of red extension occurrence

Seed	100ft	125ft	150ft	175ft	200ft
10	109	108	108	108	108
20	105	106	106	105	103
30	108	108	105	106	104
40	109	112	110	110	110
50	107	105	106	106	107
Total	**538**	**539**	**535**	**535**	**532**

Table 3 Red extension results type

Type	Description
1	Red is extended when there are red-running vehicles in the intersection. (RERV)
2	Red is not extended when there are red-running vehicles in the intersection. (RNERV)
3	Red is extended when there is no red-running vehicle in the intersection. (RENRV)
4	Red is not extended when there is no red-running vehicles in the intersection. (RNENRV)

Instead of comparing the frequency that every type of red extension occurs in each scenario, the method of comparing percentiles of every type of red extension occurs is adopted in this project. That percentile represents the probability that every type of red extension will happen in every scenario. According to table 3, when the position of detector 44 changes from 100ft to 200 ft away from the stop bar, the percentile of RERV increases from 26.4% to 39.8%; the percentile of RNENRV is opposite from RERV, which jumps from 65.2% to 49.8%; the percentile of RENRV drops from 5.9% to 2.1% when distance increases from 100 ft to 150 ft, and then begins to raise to 3.8% when distance increases from150 ft to 200 ft; the percentile of RENRV raises from 2.4% of 100 ft to 7.3% of 175 ft, and then drops to 6.6% of 200 ft. So as the distance increasing from 100 ft to 200 ft, it would appear that the system is more and more likely to extend all red because the total percentile of RERV and RENRV increases from 28.9% to 46.4%. Figure 23 is based on data of table 3, which shows the trends more readily. As shown in figure 23, the trends of RERV and RENRV are raising all the time while the distance is increasing, and RERV is turning flatten after some point; the trend of RNENRV is dropping, and will likely become flat at some point after 200 ft; the trend of RNERV drops first and slightly raise, the percentile at 200 ft is still lower than at 100 ft.

Table 4 Percentile of all red extension types

	RERV	RNERV	RENRV	RNENRV	Total
100ft	142	32	13	351	538
	26.4%	5.9%	2.4%	65.2%	100%
125ft	183	14	22	320	539
	34.0%	2.6%	4.1%	59.5%	100%
150ft	204	11	24	296	535
	38.1%	2.1%	4.5%	55.3%	100%
175ft	209	16	39	271	535
	39.1%	3.0%	7.3%	50.7%	100%
200ft	212	20	35	265	532
	39.8%	3.8%	6.6%	49.8%	100%

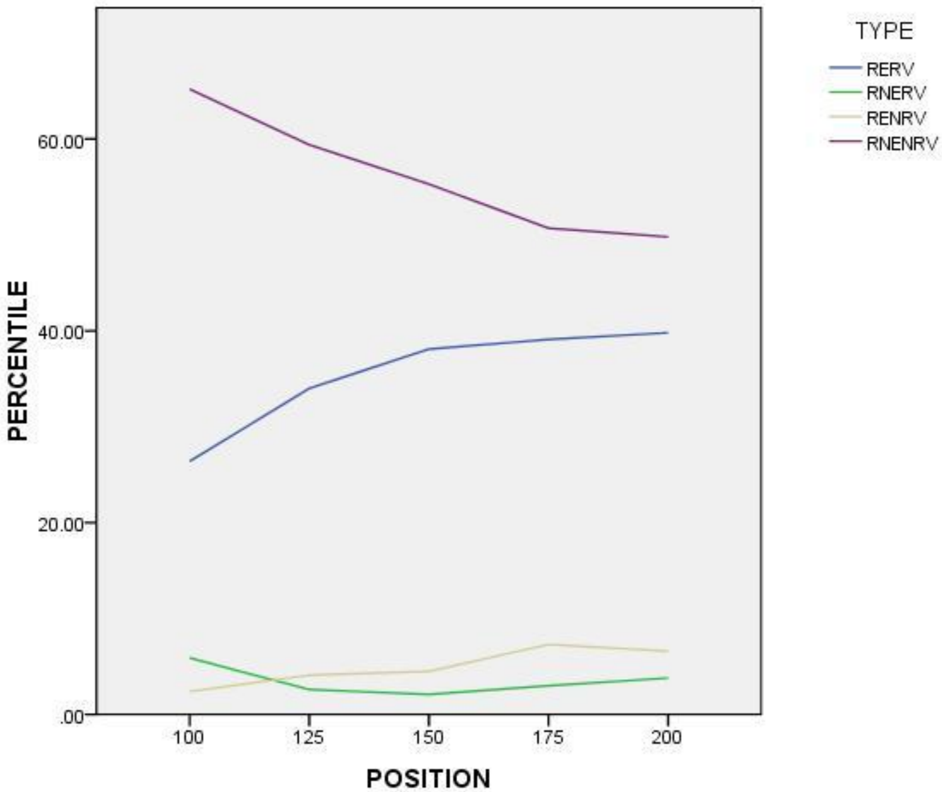


Figure 23 Position effect charts

According to the algorithm, red clearance will be extended only when vehicle's speed is larger than its stopping speed and there is not enough time for vehicle to cross over the stop bar by the end of yellow interval, in other words, the time needed tn is larger than time left of yellow interval tl. The table 4 shows the relationships between the stopping speeds and time needed of different detector locations. All data in table 5 are calculated using equation 4 and equation 5.

$$\mathrm{SSD} = \frac{V^2}{30 * (\frac{a}{32.2})}$$

Equation 4 Stopping distance

Where

SSD: stopping distance, here equals to the distance that detector locates away from stop bar

a: deceleration, $10 ft/s^2$

$$t_n = \frac{D}{V}$$

Equation 5 Time needed

Where
D: the distance that detector locates away from stop bar
V: vehicle speed

Though stopping speed is increasing while the distance of detector is increasing, the result, tn, is still increasing. Due to the increment of time needed tn, more vehicles will trigger the red extension, assuming in same condition. That's the reason that total number of RERV and RENRV is increasing.

As mentioned above, the value of time needed is increasing. Because extension time equals to time needed minus time left, it will also increase. Scatters can obviously indicates which interval extension time frequently appears within. Figure 24 shows that extension time appears within interval 0 to 1.1 second most frequently while figure 25 reflects that extension time appears within 0 to 1.5 second. According to following 5 figures, it is easy to conclude that red extension time is increasing as the increase of detector distance. Table 6 is another proof that it indicates that mean value of extension time increases as the result of maximum extension time increase.

Table 5 Stopping speed and Time needed

Position	100ft	125ft	150ft	175ft	200ft
Stopping Speed (MPH)	30.52	34.13	37.38	40.38	43.17
Time needed(Second)	3.28	3.66	4.01	4.33	4.63

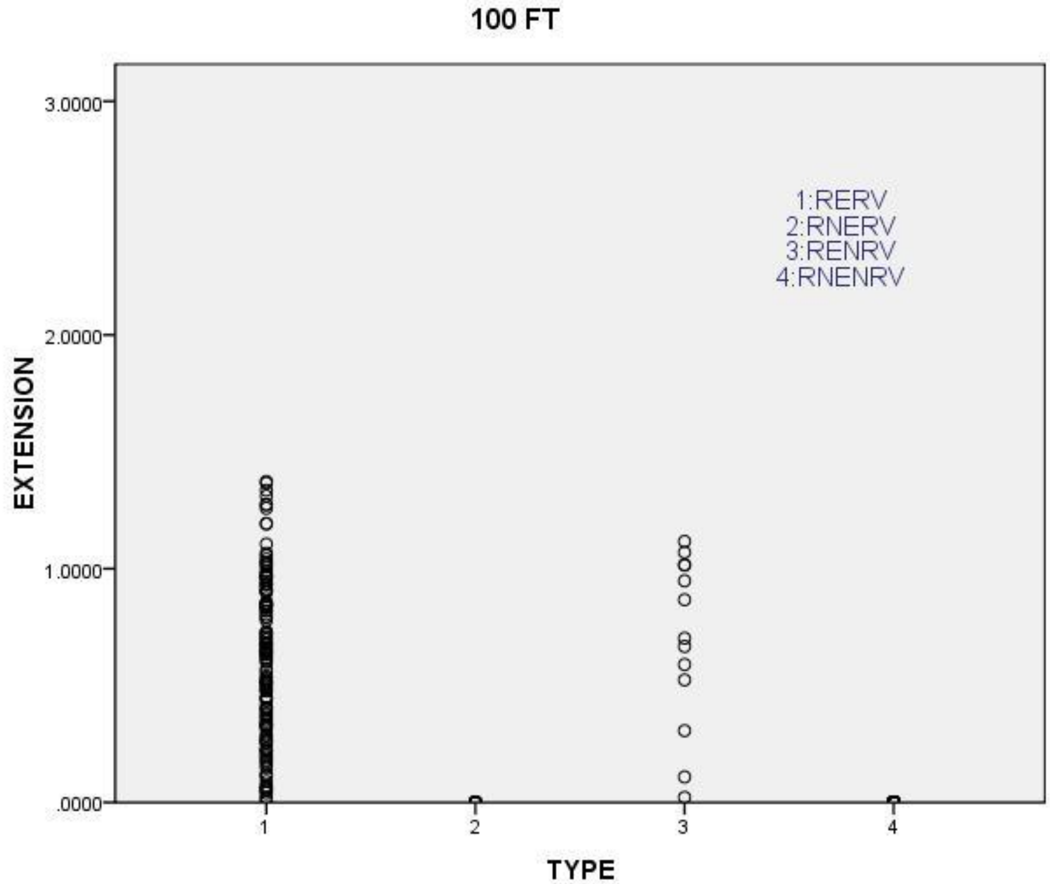


Figure 24 Scatter of position 100 feet

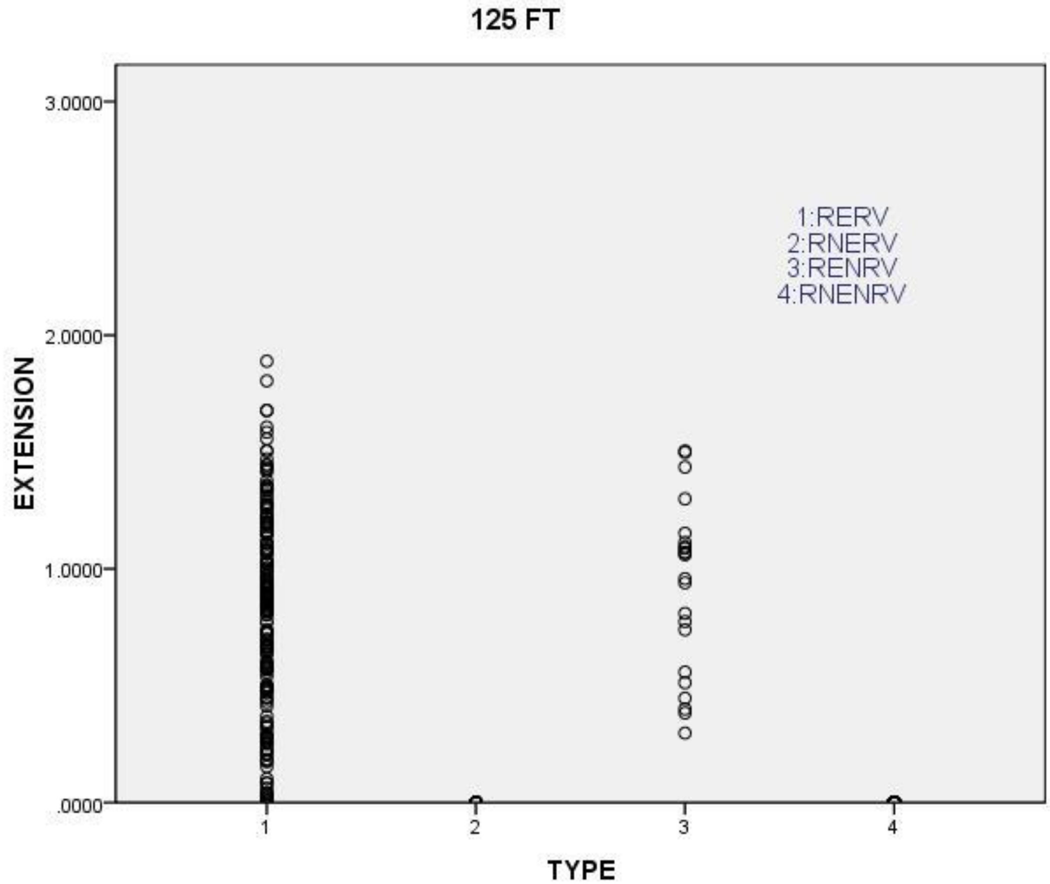


Figure 25 Scatter of position 125 feet

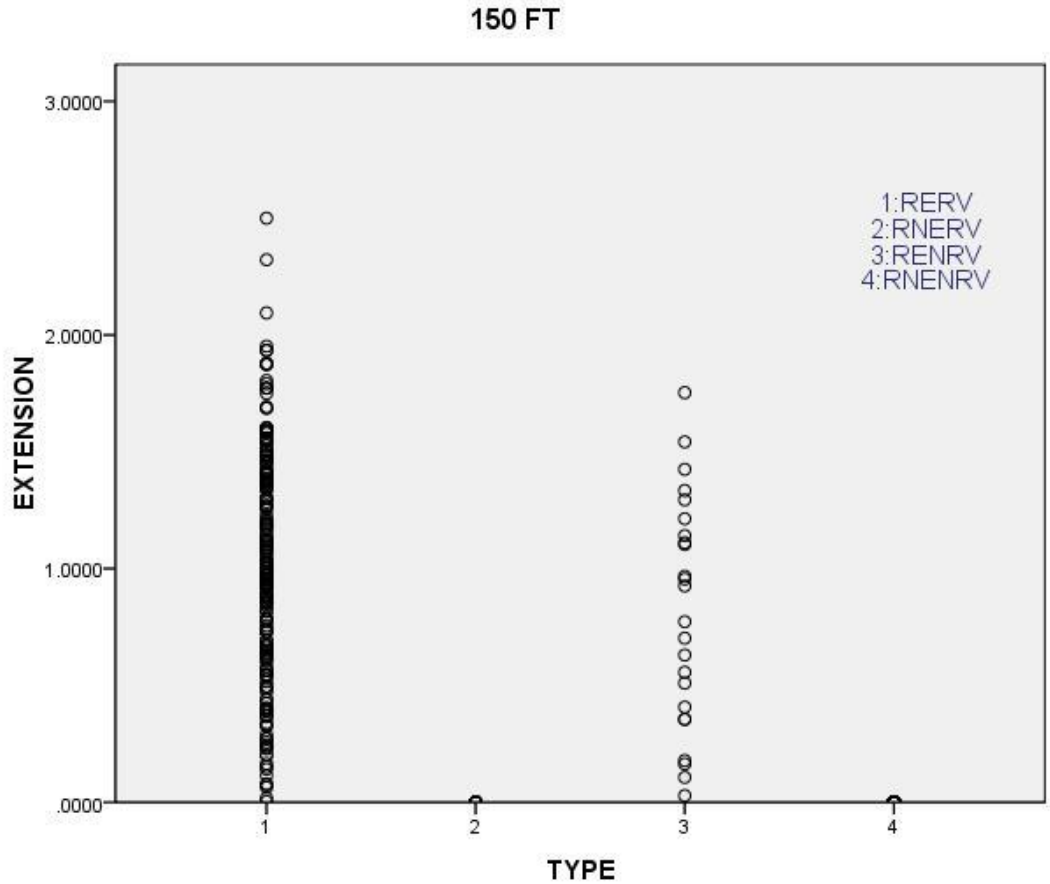


Figure 26 Scatter of position 150 feet

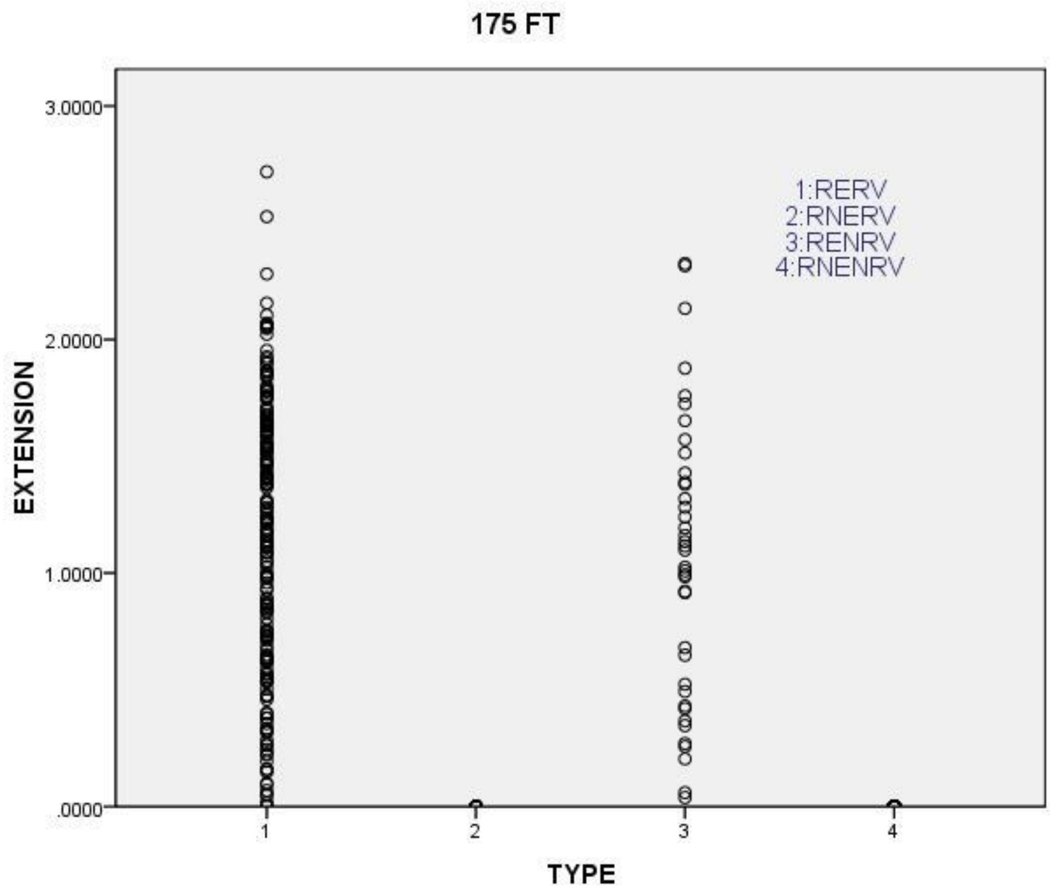


Figure 27 Scatter of position 175 feet

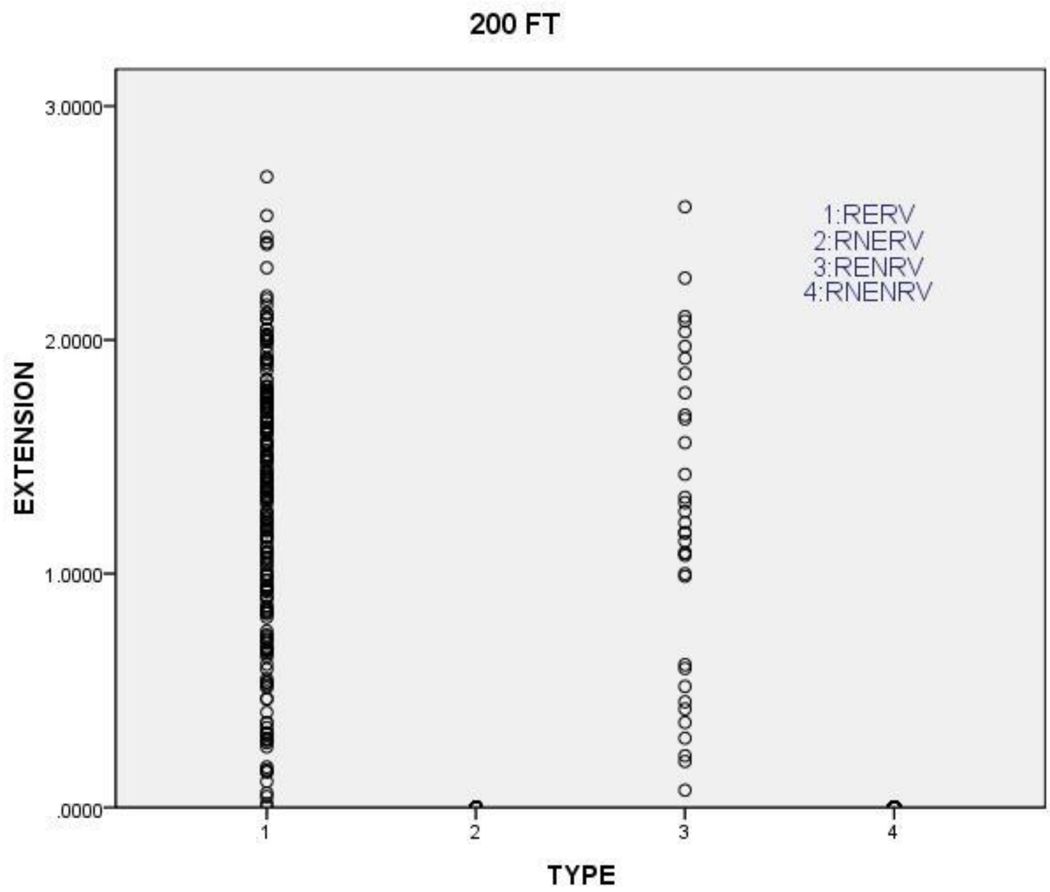


Figure 28 Scatter of position 200 feet

Table 6 Descriptive statistics

Ext time	Minimum	Maximum	Mean	Std. Deviation
100ft	0.0042	1.3734	0.6216	0.3438
125ft	0.0009	1.8889	0.8164	0.4251
150ft	0.0036	2.5000	0.9908	0.4984
175ft	0.0011	2.7182	1.1609	0.5760
200ft	0.0029	2.6980	1.2949	0.5954

CONCLUSION

From the previous data analysis, several conclusions can be gained:

First, the system is more likely to extend all red as the distance of detector increases. In order to make system more effective, it is a good approach to locate detector as far away from the stop bar as practical. But as a result of the distance increase, RENRV will be more likely to occur. RENRV is the kind of error that wastes time in every cycle, counter to this project's purpose. So the detector should not be located too far away from the stop bar;

Second, system accuracy will increase first and then decrease. Not only RENRV is the error that should be avoided, but also RNERV. Actually RNERV is far more serious than RENRV. The red clearance interval is developed to avoid accident. And the goal of this project is to improve the efficiency of it, not to create more accidents.

Therefore, the best detector location is the point with least RENRV and RNERV, with a focus on minimum RNERV. According to table 4.1.2 and figure 4.1, detector located at 150ft has the least RNERV value, 2.1%, and least summation of RNERV and RENRV, 6.6%. It can be tentatively concluded that 150 ft is the appropriate position to locate the red extension detector comparing to 100 ft, 125 ft, 175 ft and 200 ft, while the speed limit is 60 MPH.

Future works

As this study is a simplified case, there is a lot of room to improve the simulation model and procedures. Though it's simple, it provides an intial framework to execute red extension. In order to make the simulation more realistic, turning movements should be added. Additionally, multi-lanes is another aspect that should be considered. But no matter how realistic the simulation is, it's still not the same as a real field test. The driving behavior in VISSIM is not the same as the real behavior, which is more complicated. It is quite necessary to evaluate the system in field. In that case, Matlab has to talk to ASC/3 controller directly using UDP or other protocol. Also, instead of simply inquiring a detector vehicle speed, the speed has to be calculated from the time interval that the vehicle passes over two detectors. There is a problem should especially pointed out. In addition, consideration of a feedback adjustment should be used in different weather conditions. For instance, in rainy or icy day, it would be harder for vehicles to stop than in regular day. System at that time should be tuned more sensible by decreasing the trigger value of speed.

Most important, safety is always the first. Any other aspects must be of lesser importance.

LIST OF REFERENCES

1 *FHWA Data and statistics of Red light running (online)*.
http://safety.fhwa.dot.gov/intersections/col_redlight.htm#data

2 Insurance Institute for Highway Safety, *Status Report*, April 28, 2001

3 Insurance Institute for Highway Safety, Automated Enforcement Laws, April 2002

4 Collision Avoidance, Siemens Energy & Automation, Inc Intelligent Transportation Systems, 8004 Cameron Road, Austin, Texas 78754-3899, 2008

5 James A. Bonneson, Karl H. Zimmerman: Effect of Yellow-Interval Timing on the Frequency of Red-Light Violations at Urban Intersections

6 Young-Jun Moon, Kwansoo Lim, Yukyung Park: Implementation of Real-time Yellow Interval Adjustment Based on Deceleration Rates and Pavement Friction Factors at Signalized Intersections, The Korea Transport Institute

7 Sophie Midenet: Impact of the signal control strategy on red light running, INRETS-GRETIA Transport Network and Advanced Software Engineering laboratory, the French National Institute for Transport and Safety Research, 2 av General Malleret-Joinville, F-94114 Arcueil Cedex, France

8 Noor Elmitiny, Xuedong Yan, Essam Radwan, Chris Russo, Dina Nashar: Classification Analysis of Driver's Stop/Go Decision and Red-Light Running Violation, TRB 2008 Annual Meeting CD-ROM, 2007

9 Ihab El-Shawarby, Ahmed Amer, Hesham Rakha: Evaluation of Driver Stopping Behavior on High Speed Signalized Intersection Approaches, TRB 2008 Annual Meeting CD-ROM, 2007

10 Wonchul Kim, Junyi Zhang, Akimasa Fujiwara, Chang Nam Ryu, Moon Namgung: Analysis of Stopping Behavior at Urban Signalized Intersection: An Empirical Study in Korea, TRB 2008 Annual Meeting CD-ROM, 2007

11 Liping Zhang, Kun Zhou, Wei-bin Zhang and James A. Misener : Prediction of RLR from Discrete Point Detection, TRB 2009 CD-ROM, 2008

12 Vissim, Version 4.30, PTV America

13 ATACID user manual, Advanced Traffic Analysis Center, Upper Great Plains Transportation Institute, North Dakota State University

VITA

Jun Xu was born in Fuzhou, Jiangxi province, China on November 23rd 1982. He received his Bachelor degree in Electronic Engineering and Automation Science from Beijing University of Aeronautic and Astronautic in 2003. After graduation, he worked as an engineer at Jiangxi Institute of Transportation Science Research from 2003 to 2007. From August 2007, Jun Xu began to pursuit his master degree in Transportation at the Department of Civil Engineering and Environment of University of Tennessee, Knoxville. He focused on traffic safety, traffic simulation and optimization.