DEPARTMENT OF TRANSPORTATION

Development and demonstration of a novel Red Light Running Warning System using connected v2i technology

Michael Levin, Principal Investigator
Civil, Environmental, and Geo- Engineering
University of Minnesota

December 2024

Research Report
Final Report 2024-33

LRRB
LOCAL
ROAD RESEARCH
BOARD

Technical Report Documentation Page

1. Report No.	2.	3. Recipients Accession No.
MN 2024-33		

4. Title and Subtitle	5. Report Date
Development and demonstration of a novel Red Light Running Warning System using connected v2i technology	December 2024
	6.
7. Author(s)	8. Performing Organization Report No.
Michael W. Levin, Zongxuan Sun, Suiyi He, Maziar Zamanpour, Jianshe Guo	#2023005
9. Performing Organization Name and Address	10. Project/Task/Work Unit No.
Department of Civil, Environmental, and Geo- Engineering Department of Mechanical Engineering University of Minnesota 500 Pillsbury Dr SE, Minneapolis, MN 55455 111 Church St SE, Minneapolis, MN 55455	11. Contract (C) or Grant (G) No. (c) 1036342 (w0) 54
12. Sponsoring Organization Name and Address	13. Type of Report and Period Covered
Minnesota Department of Transportation Office of Research & Innovation 395 John Ireland Boulevard, MS 330 St. Paul, Minnesota 55155-1899	Final Report 14. Sponsoring Agency Code

15. Supplementary Notes
http://mdl.mndot.gov/

16. Abstract (Limit: 250 words)

Running red traffic signals is a major cause of traffic collisions and resulting injuries and fatalities. Despite extensive prior work on systems to reduce red light violations, they continue to be a major problem in practice, partly because existing systems suffer from the flaw of providing the same guidance to all drivers. As a result, some violations are avoided, but other drivers ignore or respond inappropriately to red light running systems, resulting in safety issues overall. We present a novel method of providing accurate warnings to individual drivers to avoid the broad guidance approach of most existing systems. Recognizing if a driver will run red lights is highly dependent on signal phase and timing, traffic conditions along the road, and individual driver behavior, the proposed warning system contains three parts: a traffic prediction algorithm, an individual warning signal optimizer, and a driver warning display. The traffic prediction algorithm predicts future traffic states along the road towards the signalized intersections using the latest traffic conditions obtained through vehicle-to-vehicle and vehicle-to-infrastructure communications. Then, an optimization problem is formulated to compute the optimal warning signal based on predicted traffic states and driver reaction model. Finally, the optimal warning signal is shown on the display screen to advise driver on how much braking is needed to avoid running the red light. The results of both simulated driving scenarios and real-world road tests show that the proposed system provides more effective and accurate warning signals to drivers, helping them avoid running red lights.

17. Document Analysis/Descriptors	18. Availability Statement
Red Light Running Warning System; Signalized Intersection Safety; Connected Vehicles	No restrictions. Document available from: National Technical Information Services, Alexandria, Virginia 22312

19. Security Class (this report)	20. Security Class (this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	146	

Development and demonstration of a novel Red Light Running Warning System using connected v2i technology

FINAL REPORT

Prepared by:

Michael W. Levin
Suiyi He
Maziar Zamanpour
Civil, Environmental, and Geo- Engineering
University of Minnesota

Zongxuan Sun
Jianshe Guo
Mechanical Engineering
University of Minnesota

December 2024

Published by:

Minnesota Department of Transportation
Office of Research & Innovation
395 John Ireland Boulevard, MS 330
St. Paul, Minnesota 55155-1899

Acknowledgements

We gratefully acknowledge the assistance and support of the following Technical Advisory Panel members: Sara Buermann (Wright County), Mark Callahan (Scott County), Kevin Chan (Traffic Engineering), Bradley Estochen (Ramsey County), Michael Faibanks (Metro District), Joe Gustafson (Washington County Public Works), Jackie Jiran (Research and Innovation), Jerry Kotzenmacher (Traffic Engineering), Victor Lund (St. Louis County Public Works), Steven Misgen (Metro District), Daniel Nelson (AECOM), Tim Plath (City of Eagan), Brent Rusco (Research and Innovation), Kristi Sebastian (FHWA-Minnesota Division Office), Julie Swiler (Research and Innovation), Sepehr Yadollahi (AECOM), and Susan Zarling (Traffic Engineering).

List of Abbreviations

RLRWS - Red Light Running Warning System

DSRC - Dedicated Short-Range Communications

CV - Connected Vehicle

CAV - Connected Autonomous Vehicle

SPaT - Signal Phasing and Timing

BSM - Basic Safety Message

RSU - Roadside Unit

OBU - Onboard Unit

MPC - Model Predictive Control

GUI - Graphical User Interface

CLI - Command Line Interface

Contents

List of Figures

List of Tables

Executive Summary

Red-light violations at traffic signals are a major contributor to crashes and fatalities. Right-angle type crashes typically account for most of the serious crashes at traffic signals. However, most right-angle type crashes are caused by red-light running. Many currently available vehicles include standard safety features like lane-departure warnings and brake assistance. With the emergence of connected vehicle technology, red-light running warnings could also be valuable as a standard feature to warn drivers if they are about to run a red light. An appropriately timed warning could encourage drivers to brake before they enter the intersection unsafely. Warnings could also be given in multiple forms: A "yellow" warning could alert drivers that they need to start slowing down, whereas a "red" warning might indicate that drivers should apply hard braking. Additionally, a "red" warning could be connected to the vehicle's braking system to provide automatic hard barking at the last moment if the driver fails to do so. The purpose of this project is to develop a red-light running warning system (RLRWS) and demonstrate it on Scott County's CSAH 18/CSAH 21/Southbridge Parkway intersection. The research team mainly completed the following tasks:

Task 1: The project team purchased a Roadside Unit (RSU) and worked with MnDOT to install it on Scott County's CSAH 18/CSAH 21/Southbridge Parkway intersection. This task required coordination and support from MnDOT to install a new RSU.

Task 2: The project team developed the on-road connected vehicle testbed. That involved connecting one of the existing dedicated short-range communications (DSRC) receivers to a laptop and developing an application programming interface (API) so that messages received by the DSRC receiver triggered decoding by a program on the laptop, which could then activate the RLRWS if desired. This involved setting up a RSU at a specified intersection so that Signal Phase and Timing (SPaT) information could be communicated with the DSRC receiver. The second part of this task involved connecting a GPS chip to provide positions and speeds to the API also. Finally, we connected a second GPS chip in a different vehicle to provide positions and speeds of the second vehicle over the internet or over DSRC to the laptop as well. At the end of this task, all of the information needed to run the RLRWS (SPaT, target vehicle position and speed, and the position and speed of the vehicle being followed) was available, and the operation of the program would be verified on Scott County's CSAH 18/CSAH 21/Southbridge Parkway intersection. The program would also record all data with time stamps for review. The RLRWS would later be implemented in that program in Task 6.

Task 3: The project team modified a microsimulation software to be a testbed for RLRWS. This involved modifying the control logic of a target vehicle so that it could be instructed to run red lights. The project team had prior experience working with Simulation of Urban MObility (SUMO) and it was used. We also constructed an API similar to the one in Task 2 where all necessary information for the RLRWS was made available to a single program. Within microsimulation, this could be implemented by adding an external, additional software call each time step that gathers the SPaT information and the positions/speeds of the relevant vehicles. This software call could occur as part of the target vehicle control logic modification.

Task 4: The project team developed an RLRWS for an individual target vehicle, meaning that the available information was the SPaT data and the position/speed of the target vehicle. The RLRWS was based on the stopping sight distance kinematics, but had to be calibrated to provide appropriate warnings. We generated "yellow" and "red" warnings late enough so that they did not become nuisance alarms, yet early enough so that the driver had time to stop when using the full braking authority. The RLRWS system was implemented in microsimulation and validated by watching when the system activated.

Task 5: The RLRWS system developed in Task 4 was modified with the addition of position/speed information from vehicles in front of the target vehicle. Such information could become available through vehicle connectivity or from sensors on the target vehicle. We aimed to detect a different type of red-light running scenario in which the target vehicle followed a platoon through the intersection, but only the target vehicle would run a red light. By following the platoon, the red-light violation was more likely to occur, but could be mitigated by additional warnings.

Task 6: In Task 4 and 5, we implemented RLRWS in microsimulation. Now we transfered the implementation to the connected vehicle on-road testbed developed in Task 2. Once the individual target vehicle RLRWS from Task 4 was finished in microsimulation, we started implementing it in the on-road testbed. This also involved the creation of an appropriate audio/visual warning, which was chosen by reviewing the literature on driving simulations of RLRWSs. Because we were testing the RLRWS on Scott County's CSAH 18/CSAH 21/Southbridge Parkway intersection, the warning should be sufficient to be noticed by a passenger operating the RLRWS but should not unduly distract the driver to ensure safe vehicle operations.

Task 7: Using one vehicle or a pair of vehicles, each operated by two members of the project team, we drove through Scott County's CSAH 18/CSAH 21/Southbridge Parkway intersection to demonstrate the RLRWS.

One team member will drove while a second operated the RLRWS. For a reliable on-road demonstration without running red lights, we also modified the software to display green-light "warnings", which demonstrated the ability of the system to detect signal phasing. After we validated the signal phasing, we then drove through Scott County's CSAH 18/CSAH 21/Southbridge Parkway intersection until we encountered scenarios where the RLRWS should activate. Since we followed all applicable road laws, we had to increase the sensitivity of the RLRWS to obtain consistent activation. We first demonstrated the RLRWS using the position/speed information from the target vehicle alone, and then we performed additional demonstrations of the platooning version.

Task 8: During earlier phases of the project, key benefits were selected to clearly define the benefits the state of Minnesota will receive from the results and conclusions of this research. This task produced a final memorandum that clarifies and documents the methodology used to calculate benefits, including any assumptions and steps required. In addition to quantitative calculations (when feasible), this task also included a qualitative discussion of the estimated benefits. The memorandum also included key steps that agencies could take to implement the research.

Key Findings

A RLRWS was developed and demonstrated in real-time through on-road test drives. The results showed that the proposed algorithm could provide a reliable warning signal to the driver to prevent running the red light.

Chapter 1

Introduction

Safety constitutes a paramount concern in the realm of transportation. Among the factors that contribute to severe accidents, running red lights stands out as a frequent reason. According to statistics from the Insurance Institute for Highway Safety (IIHS) [1], in the United States, 928 people were killed and an estimated 116,00 people were injured in accidents that involved red-light running in 2020. Clearly, the development of effective countermeasures targeting red-light running violations can significantly reduce the number of these accidents.

Different methods have been attempted to reduce red-light running violations at signalized intersections. These include pavement markings to indicate upcoming traffic signals [2, 3] and the use of red-light cameras as a traffic enforcement tool [4, 5]. Red-light cameras are designed to capture images of vehicles that enter an intersection after the signal has turned red, resulting in penalties for the offending drivers. Studies suggest that these cameras contribute to a 21.3% reduction in fatal collisions caused by red-light violations [6]. Despite these benefits, the deployment of automated enforcement cameras is not widespread, and their legal status varies by jurisdiction [7]. Additionally, the presence of red-light cameras can lead to unintended consequences, such as drivers braking suddenly to avoid fines, and may increase the frequency of rear-end collisions [8]. Meanwhile, frequent stops during the yellow-light phase can disrupt traffic flow efficiency at signalized intersections [9]. Similarly, using flashing green or yellow lights as a prelude to the red signal has been shown to reduce red-light violations [10, 11], but this approach also tends to increase the incidence of early stops. Countdown timers that display the time remaining before the light changes [12, 13] yield mixed results in curbing red-light violations, particularly because aggressive drivers may accelerate during the yellow phase [14]. These strategies are largely reactive, offering limited guidance to drivers on how to avoid running a red light. Therefore, it is essential to develop more proactive and effective measures to enhance safety at intersections.

One solution is to develop a red-light running warning system (RLRWS), which alerts drivers proactively to the risk of running a red light. Previous work [15, 16, 17] uses a RLRWS based on the vehicle's speed and remaining distance to the signalized intersection. However, the fast-changing traffic environment, such as the queue before the intersection, has a significant impact on the output of warning messages and should be considered in the algorithm.

Previous work shows that it is possible to predict future traffic states by using information from other connected vehicles (CVs) along with the SPaT data from traffic signals. Therefore, based on the obtained predicted traffic states, it is possible to develop a RLRWS that proactively alerts drivers to the risk of running a red light. To achieve this, we propose a new structure that calculates the optimal warning signal based on predicted traffic states from the traffic-prediction framework and the driver-reaction model. A model predictive control (MPC) problem is then formulated to determine this optimal warning signal. To experimentally validate the proposed RLRWS, a on-road testbed involving both RSU and OBU is built for this project. Test vehicles equipped with the warning algorithm are used to demonstrate the framework's performance through on-road test drives. The advantages of the proposed warning system are evaluated by assessing both the potential implementation costs, including hardware purchase and installation, and the economic benefits derived from a reduction in crashes related to red-light running violations.

This report is organized as follows: in Chapter 2, we present the hardware purchase (OBU and RSU) and installation. In Chapter 3, the configuration of the on-road connected vehicle testbed is presented. Chapter 4 talks about the development of a microsimulation testbed that will be used for RLRWS development. The algorithm design of RLRWS for an individual target vehicle and vehicle in the car-following are shown in Chapter 5 and Chapter 6, respectively. Chapter 7 presents the implementation of the proposed warning algorithm in real-time. The on-road demonstration of the proposed algorithm is summarized in Chapter 8. Chapter 9 analyzes the potential benefits of implementing the warning system. Finally, Chapter 10 presents concluding remarks and potential future work.

Chapter 2

Obtain RSU and Install it on Traffic Signal

2.1 Introduction

Safety is critical for on-road transportation. Among all factors that can result in serious accidents, red light running happens frequently and is often deadly. According to statistics from the Insurance Institute for Highway Safety (IIHS) [1], 928 people were killed in crashes that involved red light running in 2020. Additionally, an estimated 116,000 people were injured in accidents related to red light running. It's obvious that developing effective countermeasures on red light running violations can significantly reduce the number of these accidents and improve road safety. One solution is to develop a red light running warning system (RLRWS), which can provide a warning signal to drivers about the possibility of running a red light in advance.

To develop a reliable RLRWS, it's necessary to predict the vehicle's future longitudinal position. Recently, emerging connected vehicle (CV) technologies have presented a promising solution for performing short-term real-time traffic prediction [18]. Equipped with dedicated short range communication (DSRC) or cellular vehicle-to-everything (C-V2X) devices, CVs can obtain real-time traffic information such as Signal Phase and Timing (SPaT), as well as speed and location data from other CVs, via Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V) communication, respectively. The obtained traffic information can be used to estimate the current traffic state ahead of the vehicle and predict future traffic traffic states. The vehicle's future longitudinal position can then be computed based on the predicted traffic states. Therefore, it's necessary to install communication devices to enable the V2I and V2V communication for this research project. Specifically, the roadside unit (RSU) and onboard unit (OBU) are required. Since the RSUs formerly on TH-55 are no longer active, the research team purchased new devices and worked with MnDOT to install them.

2.2 Overview on RSU and OBU

The RSU and OBU are two key components of the intelligent transportation systems. The RSU is installed at the intersection, while the OBU serves as the in-vehicle equipment. The RSU and OBU can broadcast or receive messages from other devices based on SAE J2735 standard [19]. Among the different types of messages, the following are most widely used for CVs:

- Basic Safety Message (BSM): The BSM is used to make CVs exchange safety data regarding vehicles and updates at 10 Hz. The BSM contains vehicle's position information (latitude and longitude coordinates, elevation, heading angle). When the device gets access to the vehicle's Controller Area Network (CAN), vehicle's throttle, brake, transmission state and steering angle are also be available in the BSM.

- Map Data Message (MAP): The MAP message is used to convey road geographic information and updates at 1 Hz. Currently, its primary usage is to describe the signalled intersection's geography. It contains the geographic locations of the intersection's driving lanes, which describes the approach direction and connections of ingress to egress lanes. The group ID of the signal phase is also included.

- Signal Phase and Timing Message (SPaT): the SPaT message is used to convey the status of the signalized intersection and updates at 10 Hz. Along with the MAP message, SPaT message can show the state of the current signal phasing and when the next expected phase will occur.

The RSU broadcasts the signalized intersection's SPaT and MAP messages, while the OBU broadcasts the vehicle's BSM and receive messages transmitted from other CVs and intelligent infrastructures.

Different wireless communication technologies can be used for CVs communication [20],including dedicated short-range communications (DSRC), cellular network, Bluetooth, Wi-Fi, and ZigBee. Among these candidates, the DSRC and cellular network are the two most commonly used technologies in CV applications.

DSRC [21] was first introduced in the United States in 1999 and uses the 5.9 GHz frequency band. The DSRC standard is based on IEEE 802.11p [22]. The communication range between connected devices is typically up to 1000 meters.

Recently, cellular communication has also been used for CV technologies [23]. C-V2X is based on the 3rd Generation Partnership Project (3GPP) standard, and currently, 3GPP standardised 4G LTE and 5G mobile cellular connectivity are used. C-V2X includes the following two modes:

- Device-to-network: in this mode, the device is connected to the conventional cellular network for vehicle-to-network (V2N) applications such as cloud services.
- Device-to-device: in this mode, the device can communicate with other V2V or V2I devices without the use of a network. Compared to the traditional DSRC, the C-V2X can exceed the range of communication between different connected devices by about 25%.

In summary, C-V2X provides more reliable communication for intelligent transportation systems. The CVs can obtain information from other CVs and intelligent infrastructures through its device-to-device communication mode. When the device is connected to the conventional cellular network, CVs can exchange information through the cloud, which can benefit the whole on-road transportation systems. Therefore, the research team decided to purchase RSU and OBU using C-V2X standard.

2.3 Signal controller used at the target intersection

Based on the work plan, the RSU would be installed at the Scott County's CSAH 18/CSAH 21/Southbridge Parkway intersection. Therefore, the purchased device must be compatible with the traffic signal controller and cabinet used at the intersection.

The intersection uses the Cobalt traffic signal controller (see Fig. 2.1) from Econolite [24]. The device is based on National Transportation Communications for ITS (Intelligent Transportation Systems) Protocol (NTCIP), which is a family of standards used to achieve interoperability and interchangeability between computers and electronic traffic control equipment from different manufacturers. It can transfer the standardized SPaT information through the Ethernet port. Therefore, the purchased RSU must support the NTCIP interface and be able to receive the SPaT information through Ethernet communication.

Additionally, the purchased RSU should also meet other requirements, such as being able to receive the power through the socket in the traffic signal cabinet. Since the RSU would work outside, it must be able to work under various extreme weather conditions such as high winds, heavy rain, snowstorm and thunderstorm. Specifically, it should function well in the cold winter of Minnesota. The research team used all of the above criteria to select the candidate device.

2.4 Characteristic of purchased RSU and OBU

During the preparation for task 1, the project team conducted research on vendors of connected vehicle equipment and discussed the requirements for compatibility with the Cobalt traffic signal controller, hardware, and warranty support. The team finally decided to purchase the RSU [25] (Fig. 2.2) and OBU [26] (Fig. 2.3) from Commsignia

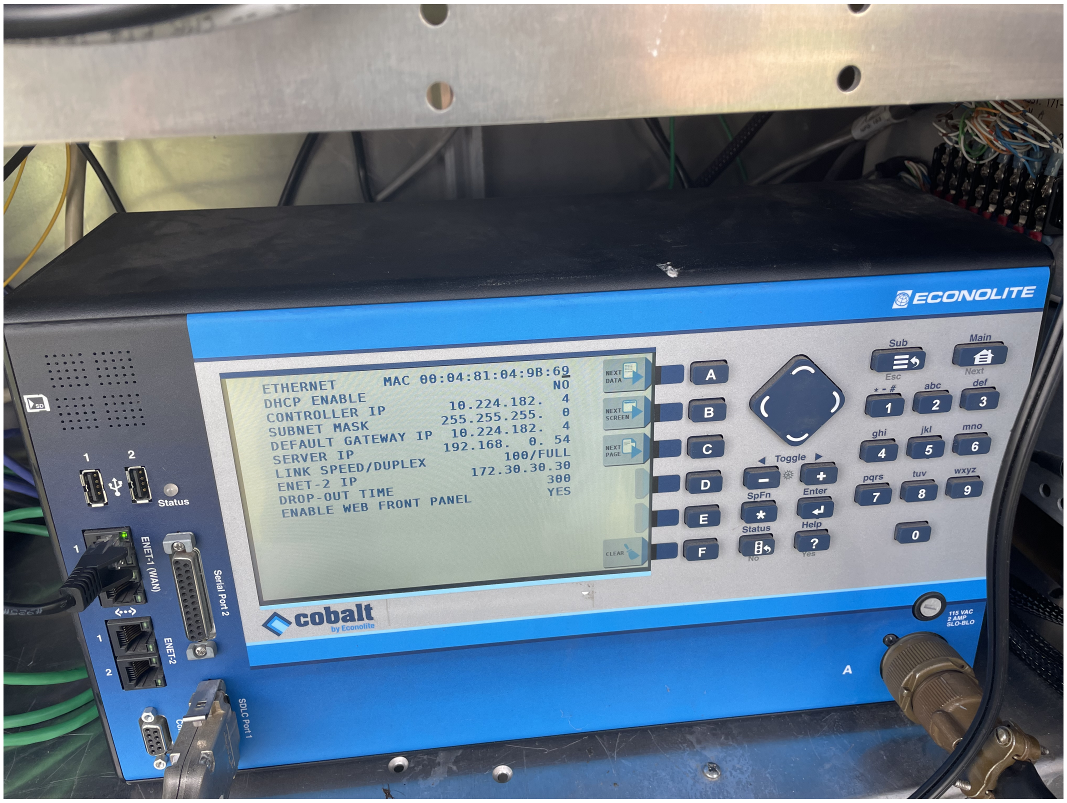


Figure 2.1: The Cobalt traffic signal controller used at the Scott County's CSAH 18/CSAH 21/Southbridge Parkway intersection.

for the project. The RSU and OBU support the C-V2X communication. Specifically, the RSU supports various traffic management center (TMC) interfaces ,which includes the NTCIP used by the Cobalt traffic signal controller. The standard power over Ethernet (PoE) works as the power supply, through which the SPaT information from the traffic signal controller can be transferred to the RSU. The device meets the requirement of NEMA4X - IP67 and can work outdoors under extreme weather conditions.

Figure 2.2: RSU with antenna.

Figure 2.3: OBU with antenna.

In addition to the basic requirements mentioned earlier, the RSU and OBU from Commsignia offer several advantages. Both devices come with a powerful onboard Central Processing Unit (CPU) and run on the Linux operating system. This allows for the development of real-time applications used on the devices. For example, the device can transmit the received messages to the laptop in real-time, which will be used for the later on-road demonstration of the red light running warning system. The devices can be accessed via Ethernet cable or Wi-Fi network, and they offer two interfaces: a graphical user interface (GUI, Fig. 2.4) and a command-line interface (CLI, Fig. 2.5). The Wi-Fi connection is particularly useful for the RSU, as it eliminates the need for a lift truck to physically connect the device to a computer for configuration changes. The manufacture also provides a V2X software development kit (SDK) in different programming languages (Python and C++), making it easy for the research team to integrate the received V2X messages with other scripts. Additionally, the manufacturer offers an online technical support team that can provide timely assistance. Based on these factors, the research team decided to purchase the RSU and OBU from Commsignia.

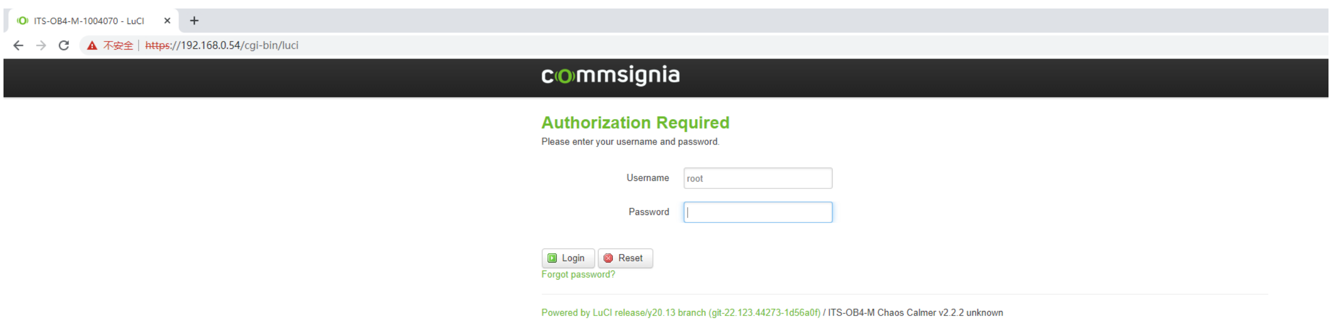


Figure 2.4: Access the device through GUI.

```
System uptime: 0 day(s) 00:07:49
V2X stack uptime: 0 day(s) 00:00:05
Memory usage: 6% of 2020MB
Device temperature: 36.0°C
Usage of /:       42% of 501.7M
Usage of /rwdata: 0% of 2.2G

RSU 4.1 operation status: Operate

IP addresses on default interface eth0
192.168.0.54
fd8f:8b8:b509:10::1


Additional hardware components:
C-V2X Modem (Qualcomm MDM9150)
LTE Modem
WiFi

C-V2X status:
HW status: OK
RX status: ACTIVE
TX status: ACTIVE
Frequency: 5915 MHz
Channel: 183
EARFCN: 55140
Bandwidth: 20 MHz
Maximum TX power: 20 dBm

root@ITS-RS4-M-1004089:~# _
```

Figure 2.5: Access the device through CLI.

2.5 RSU installation

After receiving the RSU and doing some preparation on campus, the research team came to the intersection to install the purchased RSU with the help of Scott County.

2.5.1 Install the RSU and connect it with the traffic signal controller

To ensure optimal communication range, the area between the RSU and the road section should remain free of obstructions. Therefore, the RSU was mounted to the master arm of the traffic signal (Fig. 2.6). The installation process required a lift truck and was completed in approximately 1 hour.

Figure 2.6: Mount the RSU to the master arm of the traffic signal.

The RSU is powered by the power over Ethernet (PoE), which means that the Ethernet cable will provide both power and data transmitted from the traffic signal controller. This required the Ethernet cable to be pulled from the cabinet to the master arm before the installation. A diagram of the connection between the RSU and traffic signal controller is shown in Fig. 2.7. Since the RSU was solely connected to the signal controller and did not have access to the IT fiber network, its installation posed no IT security concerns.

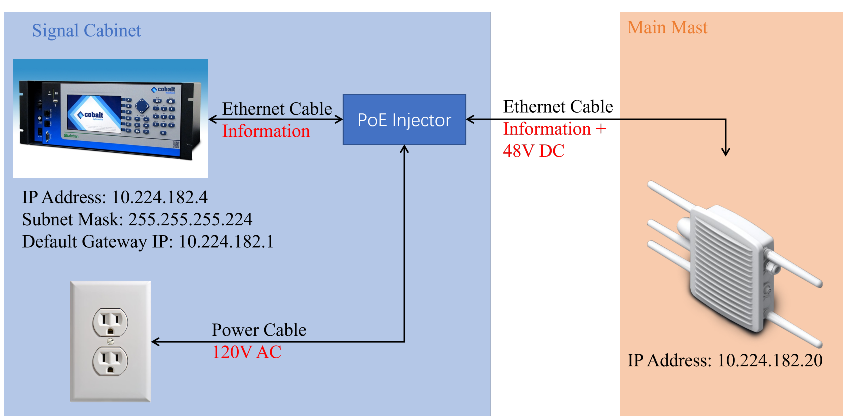


Figure 2.7: Diagram for the connection between the RSU and traffic signal controller.

2.5.2 Configuration of the traffic signal controller

The Econolite traffic signal controller supports two formats for transmitting SPaT messages to the RSU: ICD-2009 format and SAE J2735 SPaT messages format via immediate forward. For this project, SAE J2735 SPaT messages need to be transmitted to the RSU. However, the SPaT message was not enabled by default on the Cobalt traffic signal controller; the research team changed the configuration file of the traffic signal controller to enable it. Since the Cobalt controller is a simple network management protocol (SNMP) enabled network device, the MIB browser was used for this purpose. The controller's Ethernet's IP address and UDP port number were required, which were accessed through the MM-1-2-1 (Fig. 2.8) and MM-1-2-5 (Fig. 2.9) commands on the controller.

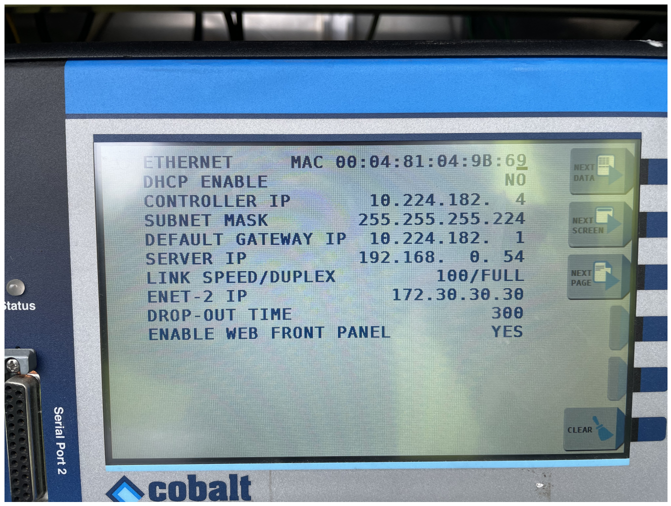


Figure 2.8: Cobalt's Ethernet Setup Menu.

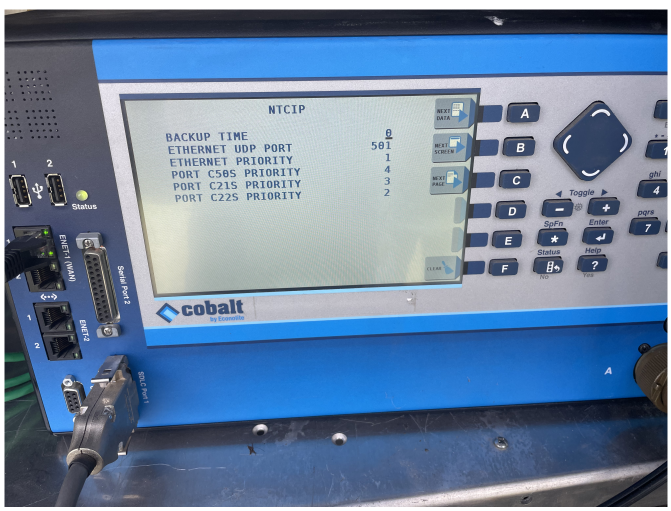


Figure 2.9: Cobalt's NTCIP Setup Menu.

Given this IP address and UDP port number, we set the MIB Browser as shown in the Fig. 2.10. Both read community and write community fields were set to public.

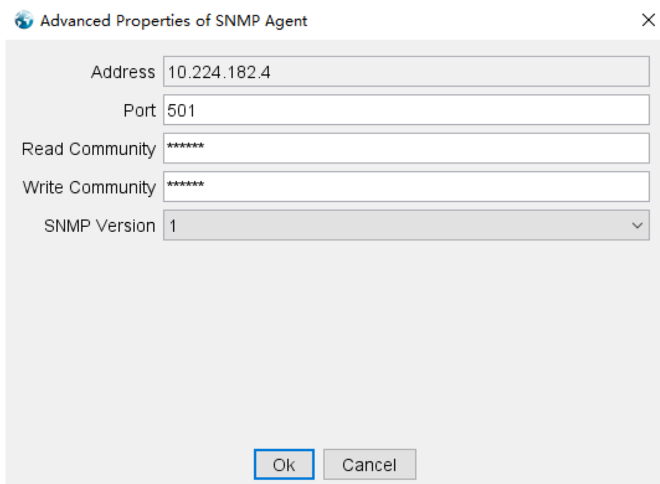


Figure 2.10: MIB Browser Setup.

Then according to the Cobalt controller manual provided by Econolit, we changed the value of object identifiers (OID) .1.3.6.1.4.1.1206.3.5.2.9.44.1.1 from 0 to 6 (Fig. 2.11), which meant that the transmitting of SPaT messages was enabled.

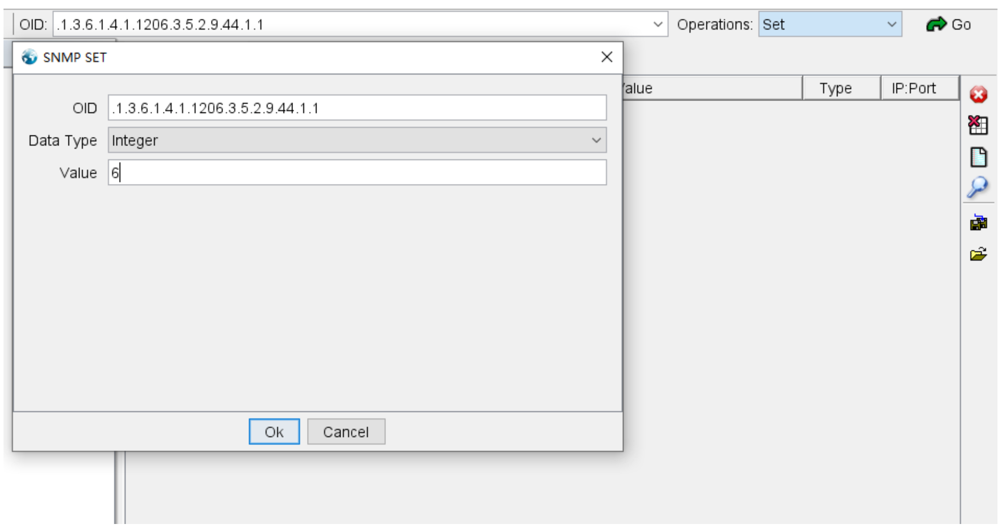


Figure 2.11: OID Field Setup.

After this, we rebooted the Cobalt controller for this change to take effect. Finally, we went to the controller's V2I/CONNECTED VEH page to set the destination IP address and port number (Fig. 2.12), which was accessed through the MM-1-2-7 command on the controller. The destination IP address was the IP address of the RSU. Since the Cobalt controller transmit the SPaT message via immediate forward, the port number was set to same as the RSU's immediate forward message UDP port. For the Commsignia's RSU, this port number is 1516.

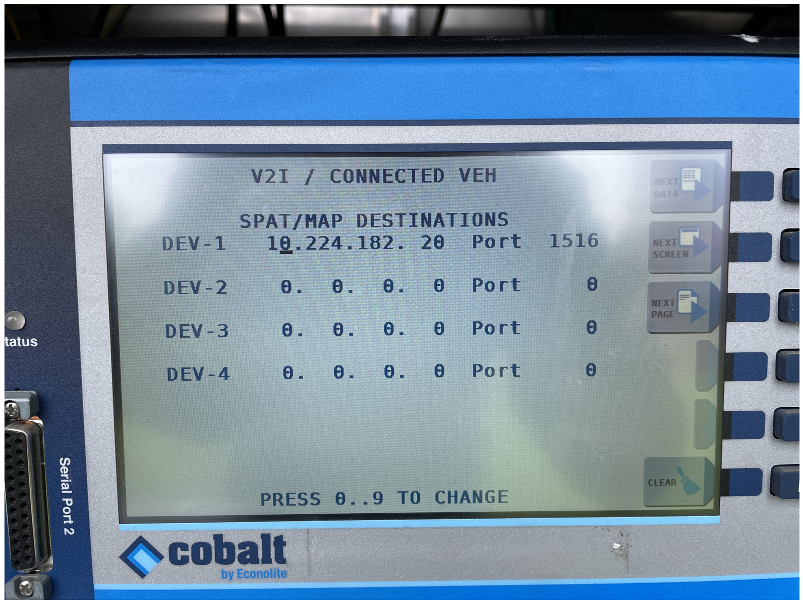


Figure 2.12: Controller V2I/CONNECTED VEH.

2.5.3 Configuration of the RSU

The first step was to save the MAP message to the RSU, which described the target intersection's geometric information and traffic signal information. The online tool [27] provided by the U.S. Department of Transportation (USDOT) was used to generate the MAP message. Firstly, a parent map was constructed, which included the intersection's geometric information such as latitude, longitude and elevation, as well as a self-defined ID number (Fig. 2.13).

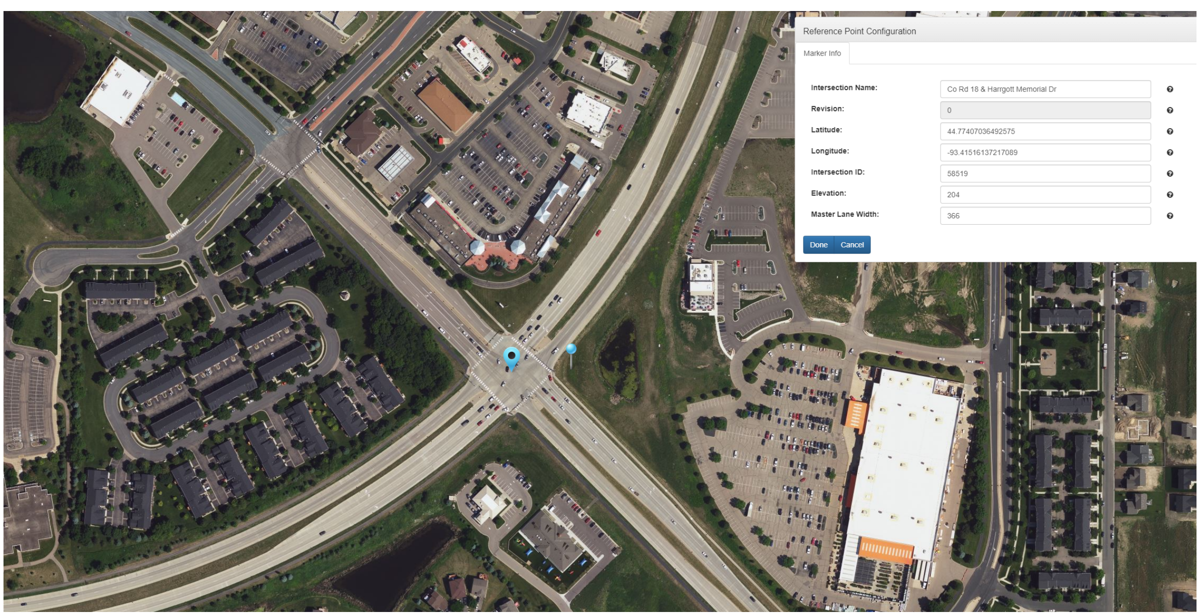


Figure 2.13: Parent Map for the MAP Message.

Then, a child map was built, which described the intersection's detailed information. Specifically, the following items were included:

- Approach: This describes the location information of the different directions of an intersection. There are two different kinds of approach: ingress approach and egress approach. Different approaches must be assigned different ID numbers.

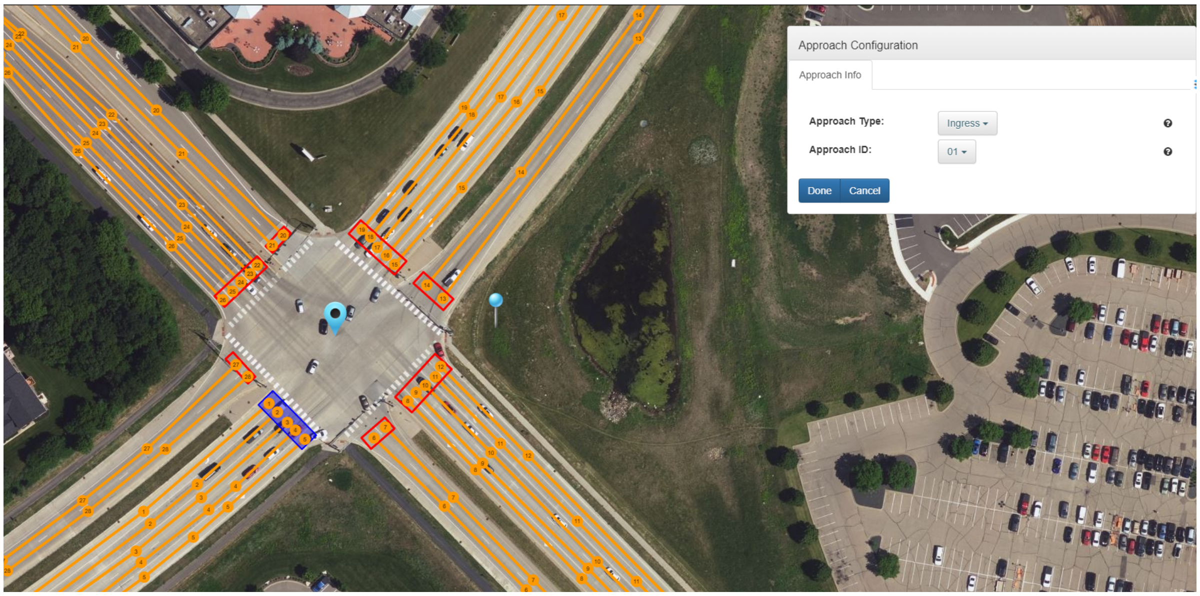


Figure 2.14: Child map with approach attributes.

- Lane: This describes the detailed geometric information of driving lanes. Each driving lane should be assigned a different ID number.

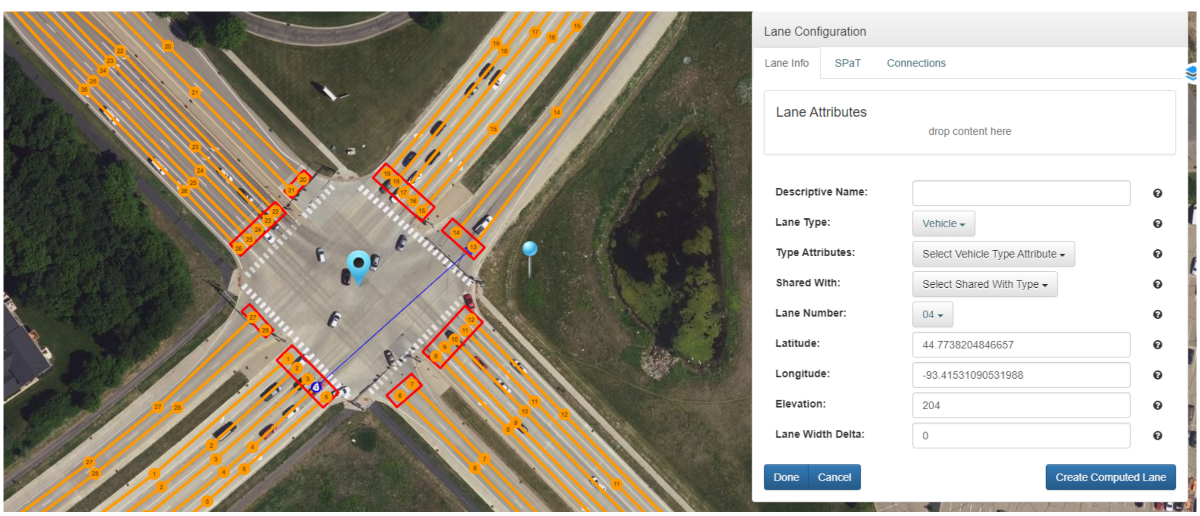


Figure 2.15: Child map with lane attributes.

- Connection: This introduces the connection relationship between the ingress lanes and egress lanes. Each connection should be assigned the correct signal group ID information, which is based on the signal controller's phase configuration (Fig. 2.17).

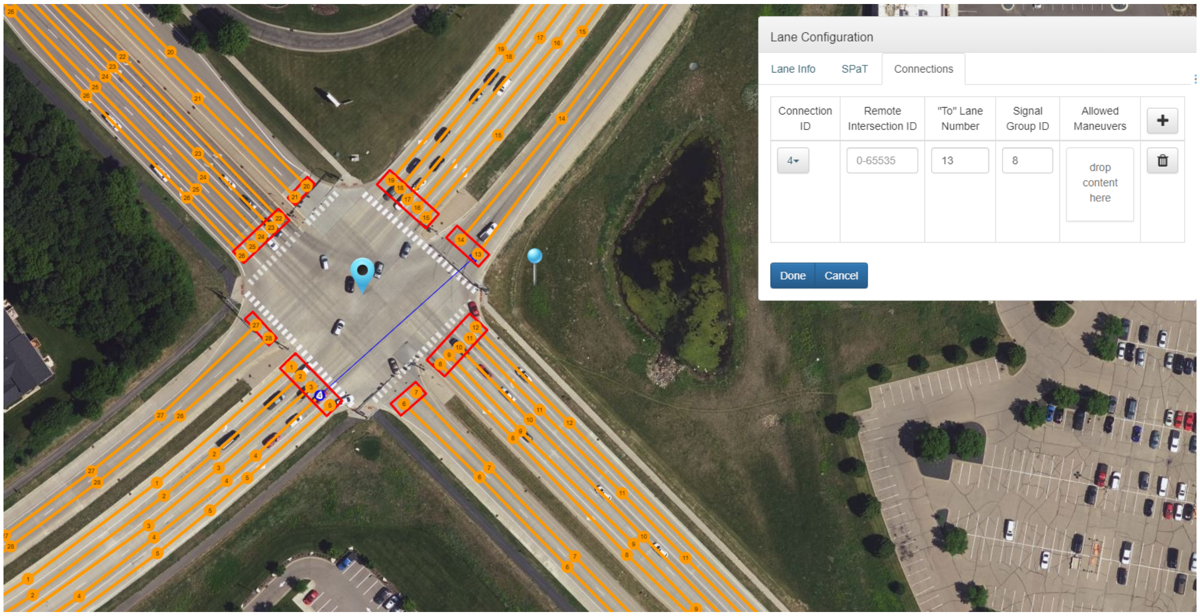


Figure 2.16: Child map with connection attributes.

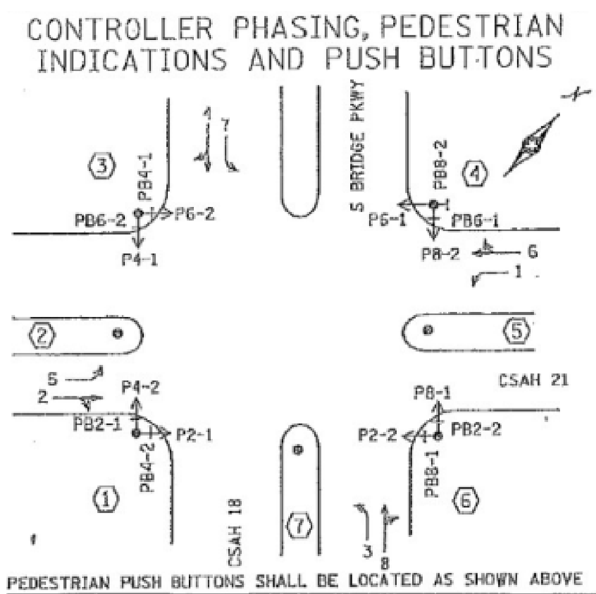


Figure 2.17: The signal controller's phase configuration at the target intersection.

Then, the generated MAP message was encoded into the UPER Hex format (Fig. 2.18)

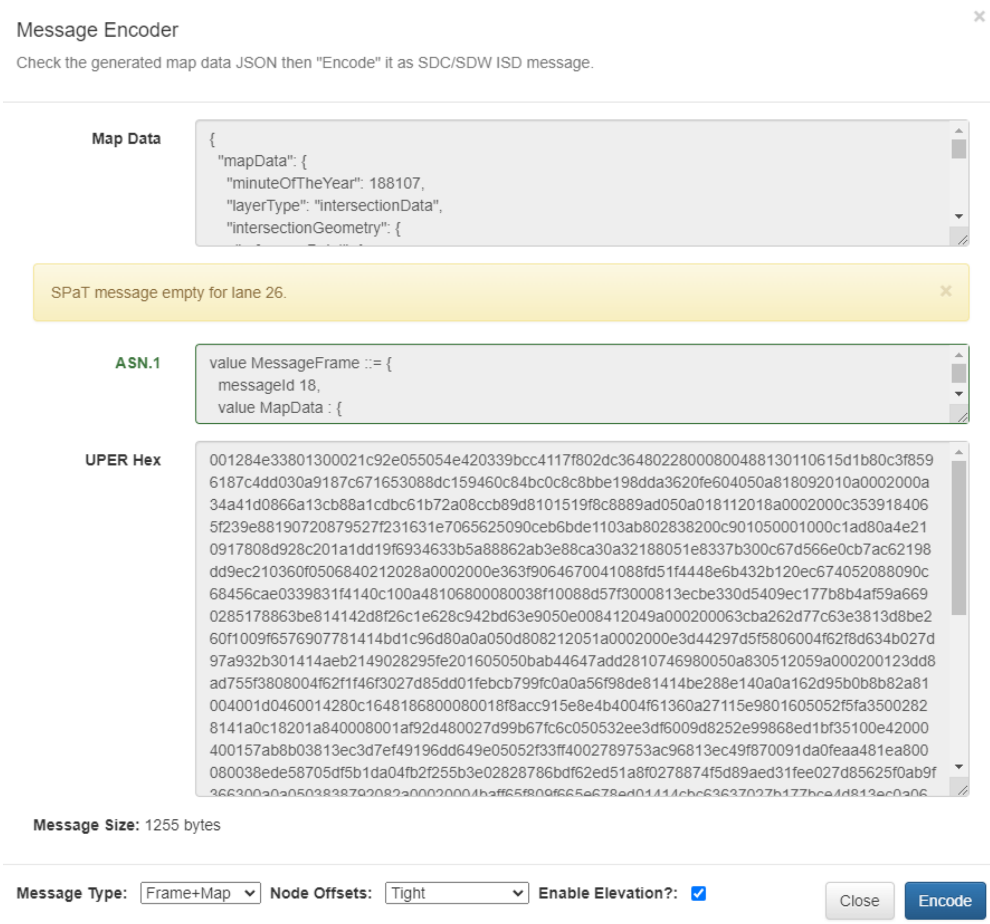


Figure 2.18: MAP Message Encoder.

To make the Commsignia RSU transmit the generated MAP message, this UPER Hex data was copied to the payload session of the message frame. This file was saved in the following directory: /rwdata/etc/rsu_msgs/. Notice that, for the Commsignia RSU, the maximum size for the MAP message is 1500 bytes. If the generated MAP message exceeds this limit, the configuration of the child map should be changed.

Since the RSU and the traffic signal controller must be in the same sub-network as shown in Fig. 2.7, we changed the RSU Ethernet port IP address from 192.168.0.54 to 10.224.182.20 (Fig. 2.19). We added the correct port number (1516 for this project) to the RSU's firewall configuration and set the listenPort of the traffic signal controller to this port number. This made the RSU listen to the SPaT message transmitted through this port number.

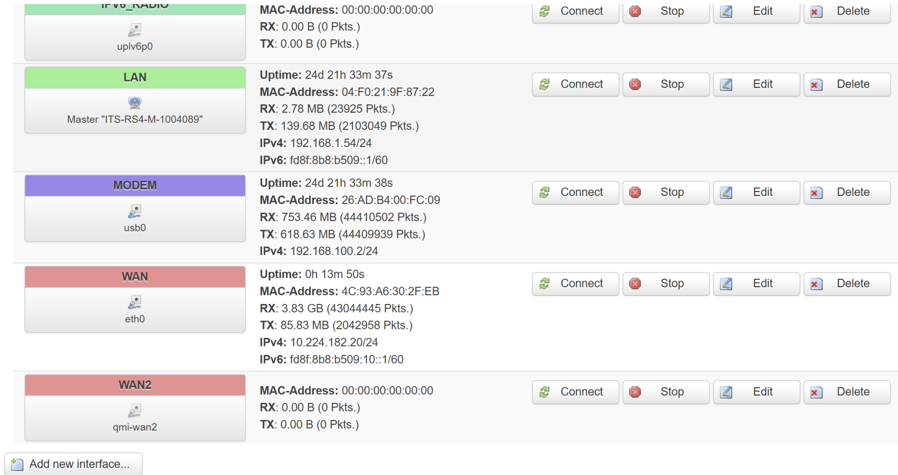


Figure 2.19: RSU Ethernet Port IP Address.

2.6 RSU and OBU communication validation

After the installation, we tried to validate the communication between the RSU and OBU. The OBU's antenna was mounted to the roof of the vehicle by magnet (Fig. 2.20). The OBU was powered by the vehicle's 12V DC power (Fig. 2.21) and was connected to the laptop through the Ethernet cable.

Figure 2.20: OBU's antenna.

Figure 2.21: OBU with 12V DC Power.

To record the received message to the OBU, the following json file was saved under the OBU's directory /rwdata/v2x_configs/data_logger_ftw:

```json
{
    "enabled": true,
    "filters": [
      {
        "direction": "In"
      },
      {
        "radioInterface": 1
      }
    ],
    "out": {
      "format": "Raw",
      "file":{
        "fileName": "/rwdata/v2x_configs/data_logger_ftw/log.pcap",
        "maxAge": 200,
        "maxSize": 600
      }
    },
    "source": [
      "Wsmp"
    ],
    "version": 6
  }
```

Figure 2.22: Json file for recording received message.

This allowed the OBU to save the received data in pcap format in the same folder. Then, Wireshark was used to open the saved data, and an online tool [28] was used to decoded the messages:

```xml
<MessageFrame>
  <messageId>20</messageId>
  <value>
    <BasicSafetyMessage>
      <coreData>
        <msgCnt>58</msgCnt>
        <id>12A7A826</id>
        <secMark>46199</secMark>
        <lat>447772478</lat>
        <long>-934145917</long>
        <elev>2030</elev>
        <accuracy>
          <semiMajor>40</semiMajor>
          <semiMinor>40</semiMinor>
          <orientation>8192</orientation>
        </accuracy>
        <transmission>
          <unavailable/>
        </transmission>
        <speed>0</speed>
        <heading>0</heading>
        <angle>127</angle>
        <accelSet>
          <long>2001</long>
          <lat>2001</lat>
          <vert>-127</vert>
          <yaw>0</yaw>
        </accelSet>
        <brakes>
          <wheelBrakes>10000</wheelBrakes>
          <traction>
            <unavailable/>
          </traction>
          <abs>
            <unavailable/>
          </abs>
          <scs>
            <unavailable/>
          </scs>
          <brakeBoost>
            <unavailable/>
          </brakeBoost>
          <auxBrakes>
            <unavailable/>
          </auxBrakes>
        </brakes>
        <size>
```

```xml
        <size>
          <width>0</width>
          <length>0</length>
        </size>
      </coreData>
      <partII>
        <SEQUENCE>
          <partII-Id>0</partII-Id>
          <partII-Value>
            <VehicleSafetyExtensions>
              <pathHistory>
                <crumbData>
                  <PathHistoryPoint>
                    <latOffset>-139</latOffset>
                    <lonOffset>-107</lonOffset>
                    <elevationOffset>-14</elevationOffset>
                    <timeOffset>5209</timeOffset>
                  </PathHistoryPoint>
                  <PathHistoryPoint>
                    <latOffset>7</latOffset>
                    <lonOffset>-65</lonOffset>
                    <elevationOffset>-17</elevationOffset>
                    <timeOffset>7210</timeOffset>
                  </PathHistoryPoint>
                  <PathHistoryPoint>
                    <latOffset>-46</latOffset>
                    <lonOffset>-31</lonOffset>
                    <elevationOffset>2</elevationOffset>
                    <timeOffset>7709</timeOffset>
                  </PathHistoryPoint>
                  <PathHistoryPoint>
                    <latOffset>-432</latOffset>
                    <lonOffset>-122</lonOffset>
                    <elevationOffset>5</elevationOffset>
                    <timeOffset>8018</timeOffset>
                  </PathHistoryPoint>
                  <PathHistoryPoint>
                    <latOffset>-442</latOffset>
                    <lonOffset>64</lonOffset>
                    <elevationOffset>6</elevationOffset>
                    <timeOffset>8069</timeOffset>
                  </PathHistoryPoint>
                </crumbData>
              </pathHistory>
              <pathPrediction>
                <radiusOfCurve>32767</radiusOfCurve>
                <confidence>200</confidence>
              </pathPrediction>
            </VehicleSafetyExtensions>
          </partII-Value>
        </SEQUENCE>
      </partII>
    </BasicSafetyMessage>
```


Figure 2.23: Received BSM message.


```xml
<MessageFrame>
  <messageId>19</messageId>
  <value>
    <SPAT>
      <timeStamp>167617</timeStamp>
      <intersections>
        <IntersectionState>
          <id>
            <id>0</id>
          </id>
          <revision>0</revision>
          <status>0000001000001000</status>
          <moy>167617</moy>
          <timeStamp>57898</timeStamp>
          <states>
            <MovementState>
              <signalGroup>1</signalGroup>
              <state-time-speed>
                <MovementEvent>
                  <eventState>
                    <stop-And-Remain/>
                  </eventState>
                  <timing>
                    <minEndTime>23047</minEndTime>
                    <maxEndTime>24176</maxEndTime>
                  </timing>
                </MovementEvent>
              </state-time-speed>
            </MovementState>
            <MovementState>
              <signalGroup>2</signalGroup>
              <state-time-speed>
                <MovementEvent>
                  <eventState>
                    <stop-And-Remain/>
                  </eventState>
                  <timing>
                    <minEndTime>22786</minEndTime>
                    <maxEndTime>22786</maxEndTime>
                  </timing>
                </MovementEvent>
              </state-time-speed>
            </MovementState>
            <MovementState>
              <signalGroup>3</signalGroup>
              <state-time-speed>
                <MovementEvent>
                  <eventState>
                    <stop-And-Remain/>
                  </eventState>
                  <timing>
```

```xml
                  </eventState>
                  <timing>
                    <minEndTime>23047</minEndTime>
                    <maxEndTime>36000</maxEndTime>
                  </timing>
                </MovementEvent>
              </state-time-speed>
            </MovementState>
            <MovementState>
              <signalGroup>4</signalGroup>
              <state-time-speed>
                <MovementEvent>
                  <eventState>
                    <stop-And-Remain/>
                  </eventState>
                  <timing>
                    <minEndTime>23047</minEndTime>
                    <maxEndTime>36000</maxEndTime>
                  </timing>
                </MovementEvent>
              </state-time-speed>
            </MovementState>
            <MovementState>
              <signalGroup>5</signalGroup>
              <state-time-speed>
                <MovementEvent>
                  <eventState>
                    <permissive-clearance/>
                  </eventState>
                  <timing>
                    <minEndTime>22798</minEndTime>
                    <maxEndTime>22798</maxEndTime>
                  </timing>
                </MovementEvent>
              </state-time-speed>
            </MovementState>
            <MovementState>
              <signalGroup>6</signalGroup>
              <state-time-speed>
                <MovementEvent>
                  <eventState>
                    <stop-And-Remain/>
                  </eventState>
                  <timing>
                    <minEndTime>22829</minEndTime>
                    <maxEndTime>22829</maxEndTime>
                  </timing>
                </MovementEvent>
              </state-time-speed>
            </MovementState>
            <MovementState>
              <signalGroup>7</signalGroup>
```

```xml
            <MovementState>
              <signalGroup>7</signalGroup>
              <state-time-speed>
                <MovementEvent>
                  <eventState>
                    <stop-And-Remain/>
                  </eventState>
                  <timing>
                    <minEndTime>23047</minEndTime>
                    <maxEndTime>36000</maxEndTime>
                  </timing>
                </MovementEvent>
              </state-time-speed>
            </MovementState>
            <MovementState>
              <signalGroup>8</signalGroup>
              <state-time-speed>
                <MovementEvent>
                  <eventState>
                    <stop-And-Remain/>
                  </eventState>
                  <timing>
                    <minEndTime>23047</minEndTime>
                    <maxEndTime>36000</maxEndTime>
                  </timing>
                </MovementEvent>
              </state-time-speed>
            </MovementState>
            <MovementState>
              <signalGroup>9</signalGroup>
              <state-time-speed>
                <MovementEvent>
                  <eventState>
                    <stop-And-Remain/>
                  </eventState>
                  <timing>
                    <minEndTime>22786</minEndTime>
                    <maxEndTime>36000</maxEndTime>
                  </timing>
                </MovementEvent>
              </state-time-speed>
            </MovementState>
            <MovementState>
              <signalGroup>10</signalGroup>
              <state-time-speed>
                <MovementEvent>
                  <eventState>
                    <stop-And-Remain/>
                  </eventState>
                  <timing>
                    <minEndTime>23047</minEndTime>
                    <maxEndTime>36000</maxEndTime>
```

```xml
                </MovementEvent>
              </state-time-speed>
            </MovementState>
            <MovementState>
              <signalGroup>14</signalGroup>
              <state-time-speed>
                <MovementEvent>
                  <eventState>
                    <stop-And-Remain/>
                  </eventState>
                  <timing>
                    <minEndTime>23047</minEndTime>
                    <maxEndTime>36000</maxEndTime>
                  </timing>
                </MovementEvent>
              </state-time-speed>
            </MovementState>
            <MovementState>
              <signalGroup>15</signalGroup>
              <state-time-speed>
                <MovementEvent>
                  <eventState>
                    <permissive-clearance/>
                  </eventState>
                  <timing>
                    <minEndTime>22798</minEndTime>
                    <maxEndTime>22798</maxEndTime>
                  </timing>
                </MovementEvent>
              </state-time-speed>
            </MovementState>
            <MovementState>
              <signalGroup>16</signalGroup>
              <state-time-speed>
                <MovementEvent>
                  <eventState>
                    <stop-And-Remain/>
                  </eventState>
                  <timing>
                    <minEndTime>23047</minEndTime>
                    <maxEndTime>36000</maxEndTime>
                  </timing>
                </MovementEvent>
              </state-time-speed>
            </MovementState>
          </states>
        </IntersectionState>
      </intersections>
    </SPAT>
  </value>
</MessageFrame>
```


Figure 2.24: Received SPaT message.

```
<MessageFrame>
  <messageId>20</messageId>
  <value>
    <BasicSafetyMessage>
      <coreData>
        <msgCnt>28</msgCnt>
        <id>12A7A839</id>
        <secMark>2599</secMark>
        <lat>447741196</lat>
        <long>-934147597</long>
        <elev>2127</elev>
        <accuracy>
          <semiMajor>40</semiMajor>
          <semiMinor>40</semiMinor>
          <orientation>8192</orientation>
        </accuracy>
        <transmission>
          <unavailable/>
        </transmission>
        <speed>0</speed>
        <heading>0</heading>
        <angle>127</angle>
        <accelSet>
          <long>2001</long>
          <lat>2001</lat>
          <vert>-127</vert>
          <yaw>0</yaw>
        </accelSet>
        <brakes>
          <wheelBrakes>10000</wheelBrakes>
          <traction>
            <unavailable/>
          </traction>
          <abs>
            <unavailable/>
          </abs>
          <scs>
            <unavailable/>
          </scs>
          <brakeBoost>
            <unavailable/>
          </brakeBoost>
          <auxBrakes>
            <unavailable/>
          </auxBrakes>
        </brakes>
        <size>
          <width>0</width>
          <length>0</length>
        </size>
      </coreData>
      <partII>
```
```
      <partII>
        <SEQUENCE>
          <partII-Id>0</partII-Id>
          <partII-Value>
            <VehicleSafetyExtensions>
              <pathHistory>
                <crumbData>
                  <PathHistoryPoint>
                    <latOffset>9</latOffset>
                    <lonOffset>-31</lonOffset>
                    <elevationOffset>1</elevationOffset>
                    <timeOffset>6749</timeOffset>
                  </PathHistoryPoint>
                  <PathHistoryPoint>
                    <latOffset>-5</latOffset>
                    <lonOffset>-37</lonOffset>
                    <elevationOffset>22</elevationOffset>
                    <timeOffset>57059</timeOffset>
                  </PathHistoryPoint>
                  <PathHistoryPoint>
                    <latOffset>93</latOffset>
                    <lonOffset>-24</lonOffset>
                    <elevationOffset>30</elevationOffset>
                    <timeOffset>60650</timeOffset>
                  </PathHistoryPoint>
                  <PathHistoryPoint>
                    <latOffset>265</latOffset>
                    <lonOffset>259</lonOffset>
                    <elevationOffset>21</elevationOffset>
                    <timeOffset>65535</timeOffset>
                  </PathHistoryPoint>
                  <PathHistoryPoint>
                    <latOffset>164</latOffset>
                    <lonOffset>459</lonOffset>
                    <elevationOffset>21</elevationOffset>
                    <timeOffset>65535</timeOffset>
                  </PathHistoryPoint>
                  <PathHistoryPoint>
                    <latOffset>110</latOffset>
                    <lonOffset>128</lonOffset>
                    <elevationOffset>9</elevationOffset>
                    <timeOffset>65535</timeOffset>
                  </PathHistoryPoint>
                  <PathHistoryPoint>
                    <latOffset>269</latOffset>
                    <lonOffset>235</lonOffset>
                    <elevationOffset>-3</elevationOffset>
                    <timeOffset>65535</timeOffset>
                  </PathHistoryPoint>
                  <PathHistoryPoint>
                    <latOffset>100</latOffset>
                    <lonOffset>45</lonOffset>
                    <elevationOffset>22</elevationOffset>
```
```
                  </PathHistoryPoint>
                  <PathHistoryPoint>
                    <latOffset>149</latOffset>
                    <lonOffset>159</lonOffset>
                    <elevationOffset>22</elevationOffset>
                    <timeOffset>65535</timeOffset>
                  </PathHistoryPoint>
                  <PathHistoryPoint>
                    <latOffset>154</latOffset>
                    <lonOffset>26</lonOffset>
                    <elevationOffset>22</elevationOffset>
                    <timeOffset>65535</timeOffset>
                  </PathHistoryPoint>
                  <PathHistoryPoint>
                    <latOffset>71</latOffset>
                    <lonOffset>-219</lonOffset>
                    <elevationOffset>31</elevationOffset>
                    <timeOffset>65535</timeOffset>
                  </PathHistoryPoint>
                  <PathHistoryPoint>
                    <latOffset>75</latOffset>
                    <lonOffset>-214</lonOffset>
                    <elevationOffset>57</elevationOffset>
                    <timeOffset>65535</timeOffset>
                  </PathHistoryPoint>
                  <PathHistoryPoint>
                    <latOffset>143</latOffset>
                    <lonOffset>-258</lonOffset>
                    <elevationOffset>52</elevationOffset>
                    <timeOffset>65535</timeOffset>
                  </PathHistoryPoint>
                  <PathHistoryPoint>
                    <latOffset>144</latOffset>
                    <lonOffset>-417</lonOffset>
                    <elevationOffset>60</elevationOffset>
                    <timeOffset>65535</timeOffset>
                  </PathHistoryPoint>
                </crumbData>
              </pathHistory>
              <pathPrediction>
                <radiusOfCurve>32767</radiusOfCurve>
                <confidence>200</confidence>
              </pathPrediction>
            </VehicleSafetyExtensions>
          </partII-Value>
        </SEQUENCE>
      </partII>
    </BasicSafetyMessage>
  </value>
</MessageFrame>
```


Figure 2.25: Received MAP message.

2.7 Summary

In this work, the research team purchased the required RSU and OBU for the red light running warning project. The RSU has been successfully installed at the target intersection and could receive the SPaT information from the traffic signal controller. The OBU was capable of receiving BSM, SPaT, and MAP messages, and these received messages could be interpreted using an online tool to extract the information for the future research project.

Chapter 3

Develop on-road connected vehicle testbed for RLRWS

June, 2023

3.1 Introduction

To develop the red light running warning system (RLRWS), the algorithm should be able to predict the vehicle's future trajectory. Since the vehicle trajectory depends on traffic condition of the road section, information from the traffic signal and other surrounding vehicles is required. For connected vehicles (CVs), the dedicated short range communication (DSRC) or cellular vehicle-to-everything (C-V2X) is used to obtain the real-time traffic information such as Signal Phase and Timing (SPaT), as well as speed and location data from other CVs, via Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V) communication, respectively.

The research team has purchased the roadside unit (RSU) and onboard unit (OBU) for Task 1 of the project. With the help of MnDOT, the RSU has been successfully installed at the intersection of Scott County's CSAH 18/CSAH 21/Southbridge Parkway. To validate the performance of the purchased RSU and OBU, the research team collected real-world traffic data in diverse traffic scenarios using these devices and analyzed the collected data. The data included CVs' trajectories, signal phase and timing (SPaT) and MAP messages.

Two different modes of data collection were employed: in the first mode, the OBU stored the CVs messages, which were then copied to the laptop for further analysis; in the second mode, the OBU broadcast the CVs messages to the laptop in real-time using the user data protocol (UDP), and the self-developed Python scripts were used to receive and interpret these messages in real-time.

Additionally, to gather information from other CVs, the OBUs from the previous project were also installed on two other test vehicles. The data were stored in the OBUs and copied to the laptop for further usage.

All the developed applications and scripts would be used for task 6: implementing RLRWS with audio/visual warning for the on-road testbed. Further details about Task 2 will be discussed in the following sections.

3.2 In-Vehicle Equipment

3.2.1 Overview

Two major in-vehicle equipment were used in this project: the OBU and the laptop. The OBU could receive different types of CV messages from various sources:

- From surrounding RSUs: This includes the SPaT messages and the MAP messages, which show the intersection's traffic signal status and geometric information.
- From other surrounding CVs: This includes the basic safety messages (BSM) from other CVs, which provide traffic information about other CVs.

Meanwhile, the OBU would also broadcast the BSM of the instrumented vehicles to RSUs and other surrounding CVs. However, since the OBUs from the previous project were based on the DSRC standard and didn't support C-V2X, they could not communicate with the newly purchased Commsignia OBU and RSU. They could only store the BSM message of the instrumented vehicles. The collected data from these older version OBUs were used solely for offline RLRWS algorithm validation. The main components of the OBU are shown in Fig. 3.1. The antenna is used to receive messages from other intelligent transportation devices (e.g. other OBU and RSU) as well as GPS signals. The power cable is used to supply power to the OBU using the vehicle's standard 12V DC power supply. The OBU processor is equipped with a powerful central processing unit (CPU) and runs the Linux operating system. It can process the received signals in real-time, store data and run user-developed scripts.

During the test drive, a laptop was connected to the OBU through an Ethernet cable. It served as the human machine interface (HMI), monitoring the status of the connected OBU, receiving and interpreting the transmitted messages. In future tasks, this laptop would also run the developed RLRWS algorithm based on the received information.

3.2.2 In-Vehicle Connection

During the on-road test, the OBU was powered by the vehicle's 12V DC power supply (green circle in Fig. 3.2). This power cable was provided by the manufacturer. To receive Global Positioning System (GPS) information and CV messages from other intelligent transportation systems, the antenna was placed on the test vehicle's roof (red circle in Fig. 3.2). The OBU processor (blue circle in Fig. 3.2) was connected to the laptop (purple circle in Fig. 3.2) using an Ethernet cable.

3.3 Data Collection

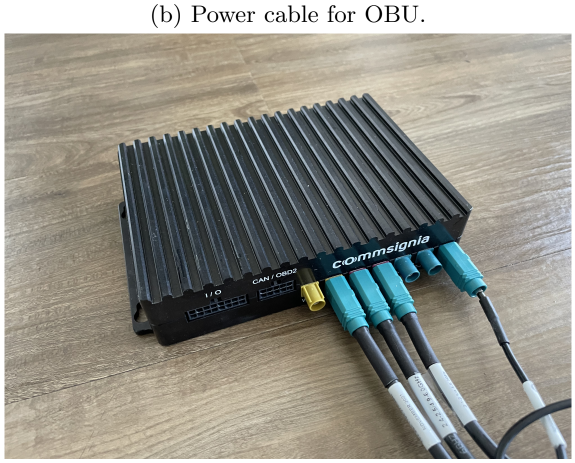
In June 2023, the research team went to the Scott County's CSAH 18/CSAH 21/Southbridge Parkway intersection multiple times to test the performance of the newly installed RSU and OBU. The team collected several groups of traffic data under different scenarios and performed analysis.

As shown in Fig. 3.3, three test vehicles equipped with OBUs were driven by research team members to collect data at the intersection. The blue vehicle was equipped with the newly purchased Commsignia OBU, which could receive both instrumented vehicle's BSM, SPaT and MAP messages from the RSU. The other two test vehicles were equipped with older version OBUs, which could not communicate with the newly purchased Commsignia OBU. They would be used to collect the instrumented vehicles' data purely.

To collect traffic data in different traffic scenarios, the project team visited the intersection during varying demand periods, such as peak and off-peak hours. Fig. 3.4 shows the traffic flow during two different time windows.

(a) GPS and C-V2X (DSRC for older version OBU) Antenna for the OBU.

(b) Power cable for OBU.



(c) OBU Processor.

Figure 3.1: Main components of the OBU are shown in the figure.

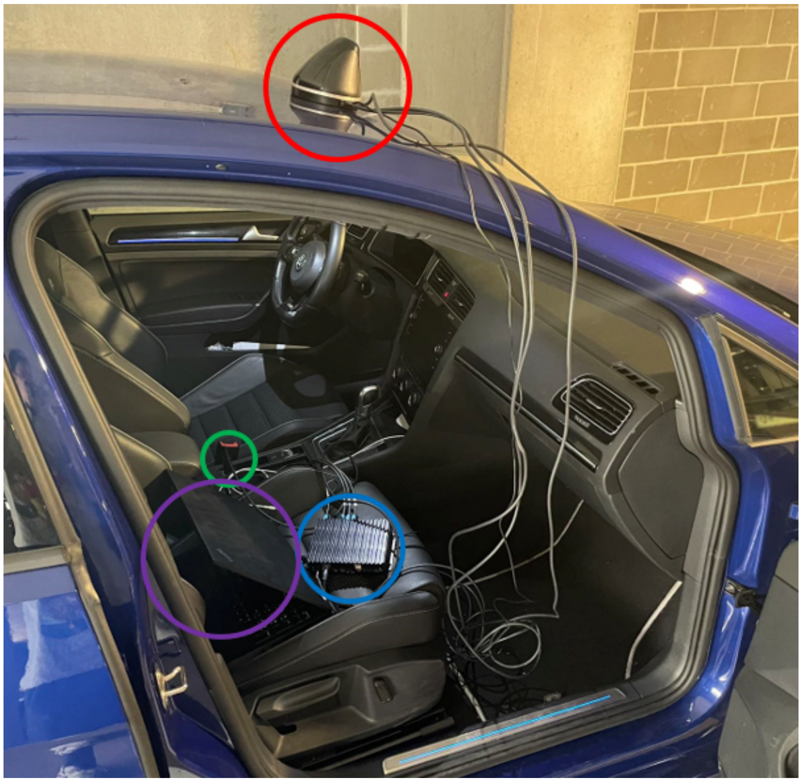
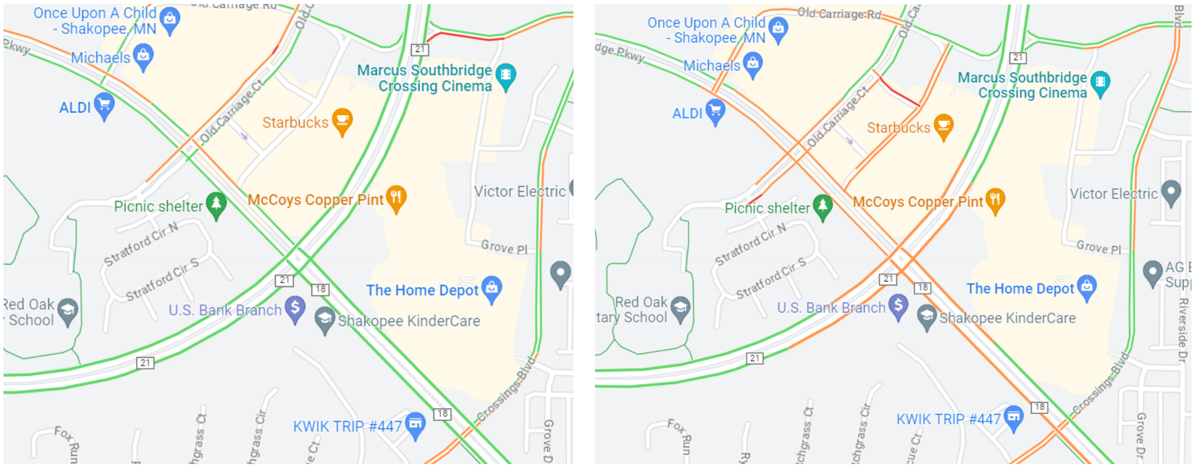


Figure 3.2: OBU installation for the test vehicle.

Figure 3.3: Test vehicles equipped with the OBU.

The green traffic flow line indicates normal traffic, the yellow traffic line shows medium traffic, and the red line indicates heavy traffic. As shown in Fig. 3.4, the intersection experienced normal traffic flow around 11:30 on weekdays and heavy traffic flow around 16:30 on weekdays. Therefore, the driving tests were conducted during both non-peak and peak times, resulting in data collection under different traffic conditions.



(a) Traffic demand at off-peak time (11:30 on weekdays)

(b) Traffic demand at peak time (16:30 on weekdays)

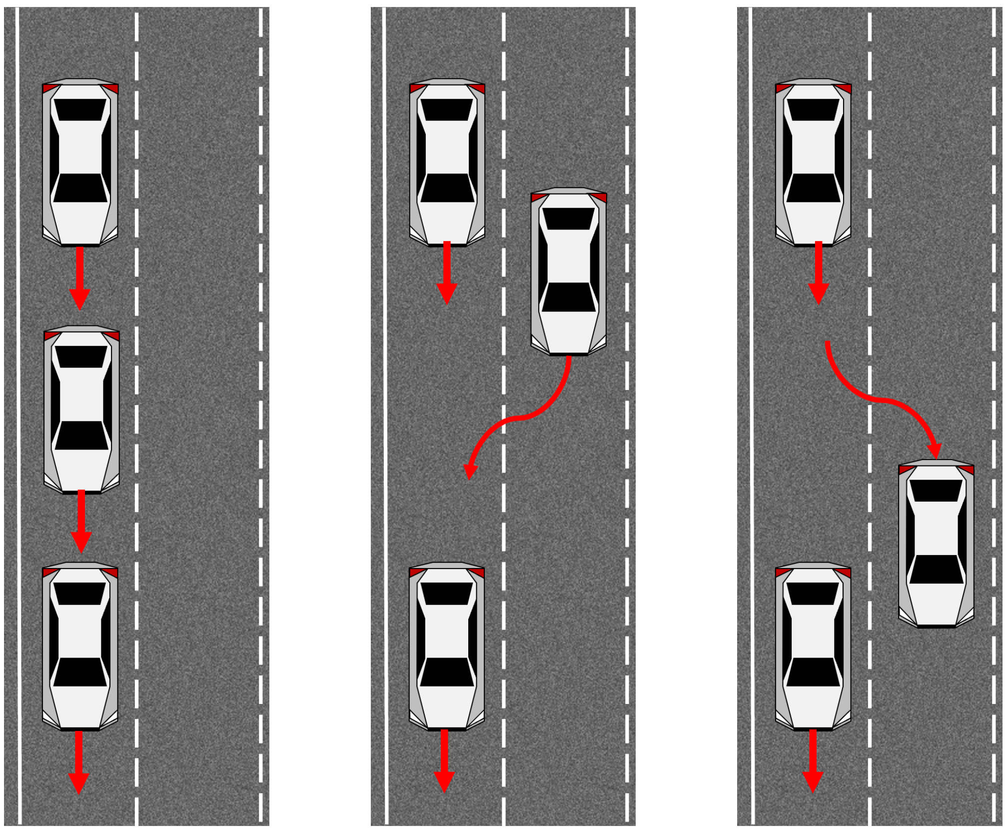
Figure 3.4: Traffic demand at the intersection.

To further capture typical driving patterns observed in the real-world, the project team defined more detailed driver behaviours during the test drives. Three different driving behaviours shown in Fig. 3.5 were implemented as the test vehicles approached the intersection: (a) car-following, (b) cut-in, (c) cut-out. The collected data were stored and would be used for analysis.

3.4 Offline Data Collection

In the first step of data collection, the collected data was stored in the OBUs, and the files were copied to the laptop for further analysis. For the newly purchased Commsignia OBU, a configuration file in JSON format was required to enable storing the collected data. Fig. 3.6 shows an example of the configuration file. The "true" variable in the line 2 is a default setting. The variables after the 'direction' attribute indicate whether the OBU's received message (with the "In" option) or transmitted message (with the "Out" option) would be stored. Similarly, the variable in the line 6 is a default setting. The "Raw" option after the 'format' attribute shows that the raw messages would be stored. The attributes 'fileName', 'maxAge' and 'maxSize' show the file's position, maximum time duration for data collection, and maximum file size, respectively. The last two attributes between the line 19 and line 22 were two variables based on the OBU's configuration version.

For the older version OBUs, no such configuration files were required. Following commands were used to start or stop recording collected data:



(a) Car-Following (b) Cut-In (c) Cut-Out

Figure 3.5: Different driver behaviours.

```json
{
    "enabled": true,
    "filters": [
        {
            "direction": "In"
        },
        {
            "radioInterface": 1
        }
    ],
    "out": {
        "format": "Raw",
        "file":{
            "fileName": "/rwdata/v2x_configs/data_logger_ftw/log.pcap",
            "maxAge": 200,
            "maxSize": 600
        }
    },
    "source": [
        "Wsmp"
    ],
    "version": 6
}
```

Figure 3.6: Configuration file in JSON format for data storage of Commsignia OBU.

- Start recording data: /etc/init.d/tcpdump start
- Stop recording data: /etc/init.d/tcpdump stop

The collected data would be stored in the folder /nojournal/pcaplogs.

Both OBUs use the packet capture (PCAP) format to save the collected data. After copying the files to the laptop, these files could be viewed using Wireshark. The online tool [28] could be used to decode the raw messages. Additionally, to interpret messages quickly, an open-source packet analyzer plugin tool [29], based on the SAE J2735 standard [19], was also employed. This tool automatically interpreted the CVs messages. An example of this tool is shown in Fig. 3.7.

3.5 Real-Time Data Collection

The newly purchased Commsignia OBU also has the capability to broadcast the received messages to the laptop in real-time. Fig. 3.8 illustrates an example of the configuration file to enable this real-time communication. The real-time communication is based on the User Datagram Protocol (UDP). Unlike the configuration file in the previous section, the destination IP address (line 14) and destination port number (line 15) are required.

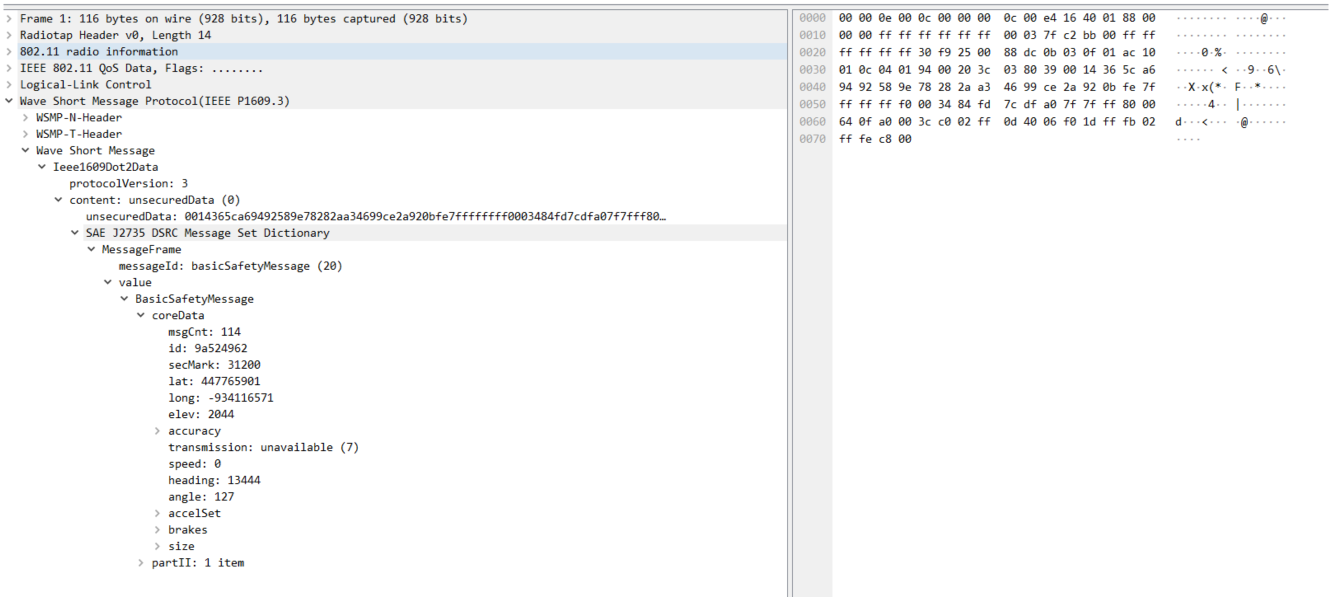


Figure 3.7: CVs messages interpretation in Wireshark.

```json
{
    "enabled": true,
    "filters": [
      {
        "direction": "In"
      },
      {
        "radioInterface": 1
      }
    ],
    "out": {
        "format": "Raw",
        "udp":{
          "destHost": "192.168.0.67",
          "destPort": 8000
        }
    },
    "source": [
      "Wsmp",
      "Gnp"
    ],
    "version": 6
  }
```

Figure 3.8: Configuration file in JSON format for real-time data transmission of Commsignia OBU.

To receive and interpret the real-time messages on the laptop, a user-developed Python scripts shown in Fig. 3.9 was utilized. The socket package was employed to receive the UDP-based data stream. The python package "pycmssdk" developed by Commsignia [30] was used to interpret the messages in real-time. In this example, the interpreted messages were displayed in the terminal. These messages would also be available for the upcoming demonstration of the RLWRS algorithm.

```
import socket
from pycmssdk import Asn1Type, FacMsgType, FacNotifData, asn1_decode, create_cms_api
from time_convert import *
red_status = ["stop-And-Remain", "stop-Then-Proceed"]
yellow_status = ["permissive-clearance", "protected-clearance", "caution-Conflicting-Traffic"]
green_status = ["permissive-Movement-Allowed", "pre-Movement", "protected-Movement-Allowed"]
error_status = ["dark", "unavailable"]
signal_id = 2  # from highway:6; to highway 2; from Home Depot 8; to Home Depot 4
UDP_IP_0 = "192.168.0.67"
UDP_PORT_0 = 8000
sock_0 = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
sock_0.bind((UDP_IP_0, UDP_PORT_0))
sock_0.setblocking(0)
UDP_IP_1 = "192.168.0.67"
UDP_PORT_1 = 7000
sock_1 = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
sock_1.bind((UDP_IP_1, UDP_PORT_1))
sock_1.setblocking(0)
while True:
    try:
        data, addr = sock_0.recvfrom(500000)
        decoded_message = asn1_decode(data, Asn1Type.US_MESSAGE_FRAME)
        if decoded_message["value"][0] == "SPAT":
            SPaT_data = decoded_message["value"][1]["intersections"][0]
            moy = SPaT_data["moy"]
            moy_hour, moy_minute = moy_to_hour_min(moy)
            time_stamp = SPaT_data["timeStamp"]
            sec_of_minute = ms_to_sec(time_stamp)
            signal_data = SPaT_data["states"][signal_id-1]["state-time-speed"][0]
            if signal_data["eventState"] in red_status:
                print("signal is red")
            elif signal_data["eventState"] in yellow_status:
                print("signal is yellow")
            elif signal_data["eventState"] in green_status:
                print("signal is green")
            min_end_time = signal_data["timing"]["minEndTime"]
            max_end_time = signal_data["timing"]["maxEndTime"]
            minute_min, sec_min = tenth_s_to_min_sec(min_end_time)
            minute_max, sec_max = tenth_s_to_min_sec(max_end_time)
            sec_to_change_min = calc_change_time(minute_min, sec_min, moy_minute, sec_of_minute)
            print("min time to change the status is after ", sec_to_change_min , " sec")
            sec_to_change_max = calc_change_time(minute_max, sec_max, moy_minute, sec_of_minute)
            print("max time to change the status is after ", sec_to_change_max , " sec")
    except BlockingIOError:
        pass
    try:
        data, addr = sock_1.recvfrom(500000)
        decoded_message = asn1_decode(data, Asn1Type.US_MESSAGE_FRAME)
        if decoded_message["value"][0] == "BasicSafetyMessage":
            latitude = decoded_message["value"][1]["coreData"]["lat"]*(10**(-7))
            longitude = decoded_message["value"][1]["coreData"]["long"]*(10**(-7))
            print('vehicle latitude',latitude)
            print('vehicle latitude',latitude)
            speed = decoded_message["value"][1]["coreData"]["speed"]*0.02
            print("Vehicle Speed: ",speed," m/s")
    except BlockingIOError:
```

Figure 3.9: Python scripts to receive the real-time UDP-based data stream.

3.6 Data Analysis

The received CVs messages were used for analysis to demonstrate the performance of the newly configured RLRWS testbed. Python scripts were written to perform this analysis.

3.6.1 SPaT

The SPaT message is employed to show the current status of a signalized intersection. The main information used in this project is the present state of the intersection and the estimated time range for its completion. Along with the MAP message, which describes the intersection's geometric layout, the instrumented vehicle can determine the state of the signal phasing and predict when the next signal status will occur. The SPaT messages are broadcast by the RSU every 0.1 seconds. The raw SPaT messages are in hexadecimal format, and the Python scripts decodes these raw messages into the SAE J2735 format. Fig. 3.10 shows an example of the interpreted SPaT messages.

From the interpreted messages, the ID of the intersection, signal phase, and timing for next signal phase are displayed. In Fig. 3.10, the data element 'id:0' represents the intersection ID, which was manually assigned in the installed RSU. This ID was used to distinguish the received SPaT messages when multiple RSUs were within the communication range. Next, the data element 'moy:218184' indicates the minute of the year, representing 151 days 12 hours and 24 minutes, equivalent to 12:24 PM on June 1st. The data element 'timeStamp: 53508' corresponds to 53.508 seconds. These two data elements show that this message was generated at 12:24:54 PM on June 1st, 2023.

The elements in 'states' show the signal status and timing of different signal phases. The data element 'signalGroup' signifies the signal phase number. Currently, the first 8 phases are used at the Scott County's CSAH 18/CSAH 21/Southbridge Parkway intersection. The data element 'eventState' displays the status of the signal. According to the SAE J2735 standard, the following states are included:

- Green Status: permissive-Movement-Allowed, pre-Movement, protected-Movement-Allowed
- Yellow Status: permissive-clearance, protected-clearance, caution-Conflicting-Traffic
- Red Status: stop-And-Remain, stop-Then-Proceed
- Error Status: dark, unavailable

```
PS G:\OBU> python .\data_processing_test.py
{'id': {'id': 0}, 'revision': 0, 'status': (520, 16), 'moy':
218184, 'timeStamp': 53508, 'states': [{'signalGroup': 1, 'st
ate-time-speed': [{'eventState': 'stop-And-Remain', 'timing':
 {'minEndTime': 15073, 'maxEndTime': 36000}}]}, {'signalGroup
': 2, 'state-time-speed': [{'eventState': 'permissive-Movemen
t-Allowed', 'timing': {'minEndTime': 15005, 'maxEndTime': 360
00}}]}, {'signalGroup': 3, 'state-time-speed': [{'eventState'
: 'stop-And-Remain', 'timing': {'minEndTime': 15073, 'maxEndT
ime': 36000}}]}, {'signalGroup': 4, 'state-time-speed': [{'ev
entState': 'stop-And-Remain', 'timing': {'minEndTime': 15073,
 'maxEndTime': 36000}}]}, {'signalGroup': 5, 'state-time-spee
d': [{'eventState': 'stop-And-Remain', 'timing': {'minEndTime
': 15073, 'maxEndTime': 36000}}]}, {'signalGroup': 6, 'state-
time-speed': [{'eventState': 'permissive-Movement-Allowed', '
timing': {'minEndTime': 15005, 'maxEndTime': 36000}}]}, {'sig
nalGroup': 7, 'state-time-speed': [{'eventState': 'stop-And-R
emain', 'timing': {'minEndTime': 15073, 'maxEndTime': 36000}}
]}, {'signalGroup': 8, 'state-time-speed': [{'eventState': 's
top-And-Remain', 'timing': {'minEndTime': 15073, 'maxEndTime'
: 36000}}]}, {'signalGroup': 9, 'state-time-speed': [{'eventS
tate': 'stop-And-Remain', 'timing': {'minEndTime': 15005, 'ma
xEndTime': 36000}}]}, {'signalGroup': 10, 'state-time-speed':
 [{'eventState': 'stop-And-Remain', 'timing': {'minEndTime':
15073, 'maxEndTime': 36000}}]}, {'signalGroup': 11, 'state-ti
me-speed': [{'eventState': 'stop-And-Remain', 'timing': {'min
EndTime': 15005, 'maxEndTime': 36000}}]}, {'signalGroup': 12,
 'state-time-speed': [{'eventState': 'stop-And-Remain', 'timi
ng': {'minEndTime': 15073, 'maxEndTime': 36000}}]}, {'signalG
roup': 13, 'state-time-speed': [{'eventState': 'stop-And-Rema
in', 'timing': {'minEndTime': 15005, 'maxEndTime': 15279}}]},
 {'signalGroup': 14, 'state-time-speed': [{'eventState': 'sto
p-And-Remain', 'timing': {'minEndTime': 15073, 'maxEndTime':
36000}}]}, {'signalGroup': 15, 'state-time-speed': [{'eventSt
ate': 'stop-And-Remain', 'timing': {'minEndTime': 15005, 'max
EndTime': 15279}}]}, {'signalGroup': 16, 'state-time-speed':
[{'eventState': 'stop-And-Remain', 'timing': {'minEndTime': 1
5073, 'maxEndTime': 36000}}]}]}
```

Figure 3.10: An example of interpreted SPaT message.

Therefore, we determined that the current signal status for signal group 1 was red. The data elements 'minEndTime' and 'maxEndTime' indicate the potential time range for the next signal status change and are in units of 1/10th of a second. Since the traffic signal controller at the intersection utilizes actuated signal control technology [31], these two data elements have different values. For instance, for signal group 1, 'minEndTime: 15073' represents 1507.3 seconds (equivalent to 25 minutes 7 seconds). This implies that the earliest possible time for signal group 1 to change to the next signal status would be at 12:25:7 PM on June 1st, 2023. The data element 'maxEndTime:36000' corresponds to 3600 seconds (equivalent to 60 minutes). This solely indicates the maximum time range for the next signal status change.

3.6.2 MAP

The MAP message is utilized to provide geometric information about the intersection, including the lane's layout, The RSU broadcasts the MAP messages every 1 second. Similar to the raw SPaT messages, the raw MAP messages are also in hexadecimal format. An example of an interpreted MAP message is presented in Fig. 3.11.

[{'id': {'id': 0}, 'revision': 1, 'refPoint': {'lat': 447740704, 'long': -934151614, 'elevation': 20
40}, 'laneWidth': 366, 'laneSet': [{'laneID': 1, 'ingressApproach': 1, 'laneAttributes': {'direction
alUse': (2, 2), 'sharedWith': (0, 10), 'laneType': ('vehicle', (0, 0))}, 'nodeList': ('nodes', [{'de
lta': ('node-XY3', {'x': -1972, 'y': -1980})}, {'delta': ('node-XY4', {'x': -3747, 'y': -3216})}, {'
delta': ('node-XY4', {'x': -3588, 'y': -2666})}, {'delta': ('node-XY4', {'x': -3599, 'y': -2328})},
{'delta': ('node-XY4', {'x': -3758, 'y': -2529})}, {'delta': ('node-XY4', {'x': -2282, 'y': -1439})}
, {'delta': ('node-XY3', {'x': -1168, 'y': -667})}, {'delta': ('node-XY4', {'x': -3896, 'y': -1672})
}, {'delta': ('node-XY4', {'x': -2972, 'y': -1090})}, {'delta': ('node-XY4', {'x': -2505, 'y': -741}
)}, {'delta': ('node-XY3', {'x': -1794, 'y': -508})}]), 'connectsTo': [{'connectingLane': {'lane': 2
1}, 'signalGroup': 3, 'connectionID': 1}]}, {'laneID': 2, 'ingressApproach': 1, 'laneAttributes': {'

Figure 3.11: An example of interpreted MAP message.

The following key data elements are crucial for CV applications. The data element 'id:0' represents the intersection ID, which is used to match the appropriate SPaT message. The 'refPoint' contains the latitude, longitude and elevation of a reference point, which will be used to compute the offset of subsequent data points. The data element 'laneWidth' denotes the driving lane's width, measured in centimeters. The data element 'laneID' indicates the assigned index for the driving lane. The data element 'ingressApproach' signifies that the lane is an ingress lane. The data element 'nodeList' is used to describe a list of positions for the lane's nodes. Each node has an X and Y offset from the previous node point. As per the definition, the X-axis extends from west to east and the Y axis extends from south to north. The units are in centimeters. The lane's node list consists of a sequence of points that describe the lane's centerline. Finally, the connection between two lanes and signal phase information are displayed in the data element 'connectTo'.

3.6.3 BSM

The BSM message is used to exchange safety data concerning the vehicle's state, including position and speed information. This data will be used in this project. The OBU broadcasts the BSM messages every 0.1 seconds. The raw BSM messages are also in hexadecimal format. An example of an interpreted BSM message is shown in Fig. 3.12.

```
{'coreData': {'msgCnt': 117, 'id': b'\x12\xa7\xa8&', 'secMark': 45100, 'lat': 447767002, 'long': -93
4116179, 'elev': 2029, 'accuracy': {'semiMajor': 40, 'semiMinor': 40, 'orientation': 0}, 'transmissi
on': 'unavailable', 'speed': 58, 'heading': 21872, 'angle': 127, 'accelSet': {'long': 2001, 'lat': 2
001, 'vert': -127, 'yaw': 0}, 'brakes': {'wheelBrakes': (16, 5), 'traction': 'unavailable', 'abs': '
unavailable', 'scs': 'unavailable', 'brakeBoost': 'unavailable', 'auxBrakes': 'unavailable'}, 'size'
: {'width': 0, 'length': 0}}, 'partII': [{'partII-Id': 0, 'partII-Value': ('VehicleSafetyExtensions'
, {'pathHistory': {'crumbData': [{'latOffset': -10, 'lonOffset': -694, 'elevationOffset': 1, 'timeOf
fset': 36180}, {'latOffset': 468, 'lonOffset': -2842, 'elevationOffset': 0, 'timeOffset': 36580}, {'
latOffset': 148, 'lonOffset': -3628, 'elevationOffset': 1, 'timeOffset': 36750}, {'latOffset': -1492
, 'lonOffset': -4216, 'elevationOffset': 3, 'timeOffset': 37090}, {'latOffset': -4203, 'lonOffset':
-6146, 'elevationOffset': 9, 'timeOffset': 37620}, {'latOffset': -5008, 'lonOffset': -6118, 'elevati
onOffset': 10, 'timeOffset': 37790}, {'latOffset': -5632, 'lonOffset': -5243, 'elevationOffset': 8,
'timeOffset': 37970}, {'latOffset': -5848, 'lonOffset': -3329, 'elevationOffset': 6, 'timeOffset': 3
8220}, {'latOffset': -6757, 'lonOffset': 6014, 'elevationOffset': 6, 'timeOffset': 39020}, {'latOffs
et': -6514, 'lonOffset': 7534, 'elevationOffset': 6, 'timeOffset': 39190}, {'latOffset': -5732, 'lon
Offset': 8704, 'elevationOffset': 3, 'timeOffset': 39360}]}, 'pathPrediction': {'radiusOfCurve': 327
67, 'confidence': 20}})}]}
```


Figure 3.12: An example of interpreted BSM message.

The data elements 'msgCnt' and 'id' represent the message ID information. The 'secMark' denotes the milliseconds within a minute, indicating the time at which the message was generated. The data elements 'lat', 'long' and 'elev' show the vehicle's latitude, longitude and elevation based on GPS signal, respectively. The elevation is measured in units of 0.1 meters. The data element 'speed' indicates the vehicle's speed, measured in units of 0.02 m/s.

3.6.4 Communication Reliability

The performance and reliability of the installed RSU were evaluated through driving tests. Since driving towards the intersection from the north towards the intersection required accessing the highway, the project team conducted driving tests from other directions. The research team initially examined the maximum and effective range of RSU-transmitted message reception. Theoretically, the communication range of intelligent transportation systems can reach up to 1000m. However, the actual range of communication depends on factors such as the geographic location of traffic signals and roadway, obstructions from stationary objects (e.g. building), moving objects (e.g. vehicles), the position of the OBU antenna, and weather conditions [32]. The collected data revealed that the communication range varied for different driving directions. The red points in Fig. 3.13 represent the positions where the OBU successfully received the SPaT and MAP messages from the RSU. When the test vehicles approached the intersection on County Road 21 from the south, they could receive the SPaT information at a range of approximately 500m ahead of the intersection.

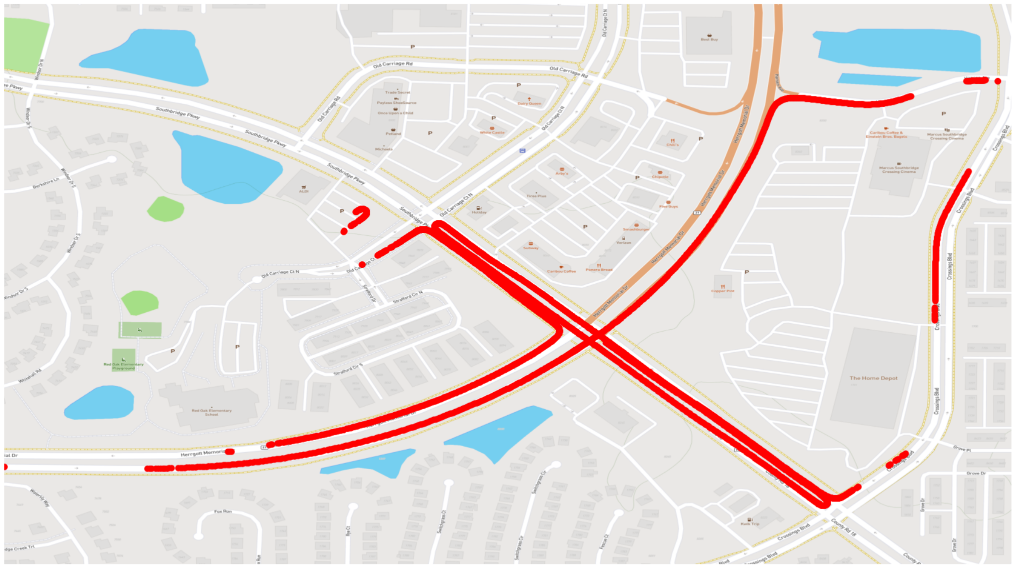


Figure 3.13: Communication reliability of the installed RSU.

3.6.5 Test Vehicles Trajectory and SPaT

Fig. 3.14 shows the vehicle trajectories of test vehicles along with the signal phase and timing at the intersection when the vehicles were traveling on the County Road 21 in the northbound direction. The blue, orange, and purple lines represent the vehicle trajectories of three test vehicles. The middle line indicates the traffic signal status at the intersection. It is evident that all test vehicles came to a stop at the intersection when the light was red and proceeded through the intersection once the light changed to green.

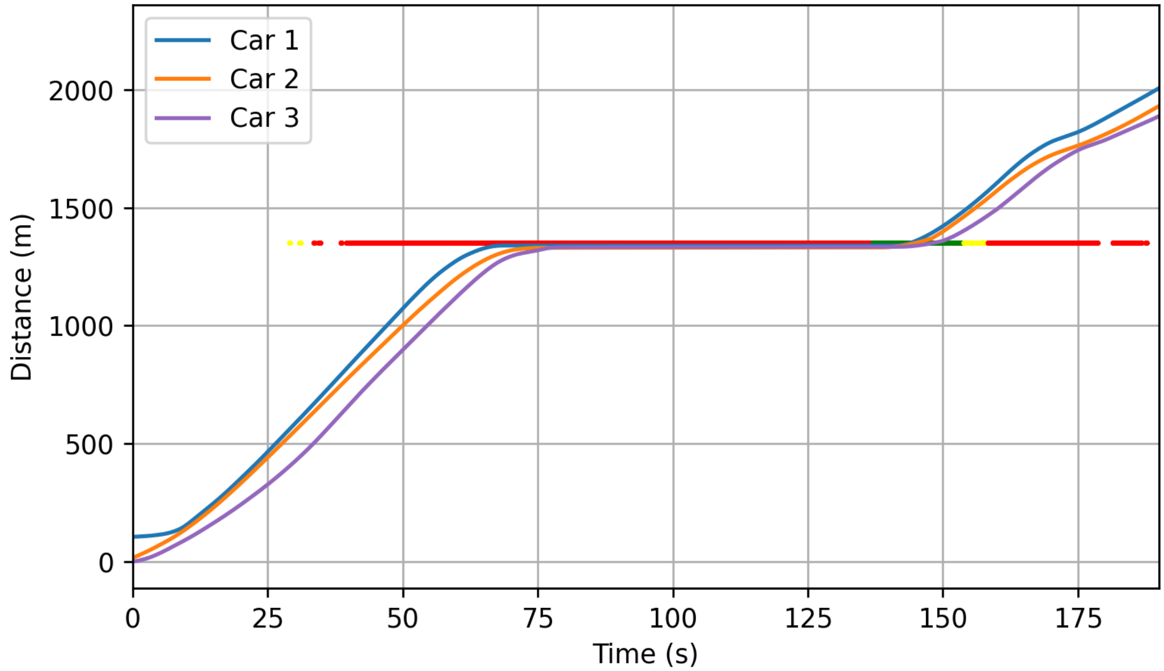


Figure 3.14: Test vehicles trajectories and SPaT information.

3.7 Summary

In this study, real-world traffic data was obtained under different traffic conditions using OBUs and RSU. The traffic data comprised information from the test vehicles as well as the signal phase and timing at the intersection. The research team developed algorithms to read and analysed data, which would be utilized for the development of RLRWS. The results show that the newly installed OBUs and RSU function properly. The RSU successfully received accurate SPaT information from the traffic signal controller, while the OBU received the SPaT messages broadcasted by the RSU. Furthermore, the real-time communication between the OBU and laptop was also validated during this task, serving as preparation for the RLRWS demonstration.

Chapter 4

Develop microsimulation testbed for RLRWS

April, 2023

4.1 Introduction

Connectivity brought a novel opportunity into traffic engineering. Connected vehicles are able to transfer data between each other and even to the signal. The current task will use the connection between a target vehicle and signalized intersection. The main goal of this task is not how to find and recognize RLRWS (Red Light Running Warning System); this process will be developed in further tasks. However, it is required to generate a testbed and demonstrate this phenomena. The testbed for RLRWS (Red Light Running Warning System) has been created by modifying through a microsimulation. Specifically, the control logic of a target vehicle has been adjusted to enable it to run red lights. Given the project team's familiarity with VISSIM, Aimsun, and SUMO (Simulation of Urban Mobility), one of these softwares is used, although the choice is not critical for testing the RLRWS as the goal is to implement the illegal behavior of running red lights. Additionally, an API similar to the one developed in Task 2 will be constructed to provide all the necessary information for the RLRWS to a single program. Within microsimulation, this will involve adding an external software call at each time step that collects SPaT information and the positions/speeds of relevant vehicles. This call will be incorporated as part of the target vehicle control logic modification.

Generally, the objective of Task 3 is to adjust the control logic of a target vehicle to enable it to run red lights, with a focus on analyzing both normal driver behavior and red light runner behavior. By default, simulation platforms are equipped with strict safety protocols to ensure that vehicles abide by driving rules and prevent accidents. As a result, these platforms do not allow vehicles to violate traffic rules, including running red lights. However, it is possible to modify the simulation's default settings to allow for the demonstration of red light running by programming the simulation to adjust its safety protocols when detecting certain behaviors exhibited by red light runners. While drivers typically stop at red signals and wait for them to turn green, some drivers disregard red signals and proceed through intersections. Task 3 aims to address both types of behavior, with an emphasis on capturing the behavior of red light runners.

The project team aims to develop techniques to demonstrate red light running at the intersection under consideration. The first step involves selecting a vehicle that will be programmed to run the red light (demonstrate RLR), and the team will ensure that the simulation process can accurately represent this. These methods must be compatible with the simulation process and feasible to implement. Once the team has developed a set of viable methods, the best option is selected for implementation on the microsimulation platform.

Assuming that the algorithm for detecting and monitoring vehicles has been finalized from other tasks, the simulation system will generate responses based on the inputs it receives. These responses may prompt the system to provide specific instructions to the vehicle, such as "keep going," "be careful," or "apply emergency brakes." If the response calls for a warning or emergency brake, the simulator will take control of the vehicle and initiate a potential stop before passing through the intersection.

4.2 Platform for microsimulation

The project team has selected SUMO for microsimulation in this project. SUMO (Simulation of Urban Mobility) is an open-source traffic simulation software developed by the German Aerospace Center (DLR). It is designed to simulate real-world traffic scenarios and the project team has prior experience of working with it.

SUMO is capable of simulating a range of transport modes, including cars, buses, bicycles, and pedestrians, and can model complex interactions between these modes. It can use advanced traffic flow models to simulate the behavior of vehicles and other road users, such as their acceleration, deceleration, and lane-changing behavior, and can simulate a range of traffic control systems, including traffic lights and roundabouts. It is possible to monitor a certain vehicle and change values for its states such as speed.

There are several advantages for using SUMO for traffic simulation. SUMO is open source, performs comprehensive simulation, has realistic behavior, integrates with other tools, is user-friendly and the most significant, it is controllable and programmable. Users can implement what they mean by this feature.

The SUMO simulation provides the ability to model a variety of networks and traffic scenarios, including traffic demand, signal properties, and lane and route details. For the purposes of this study, the project team will be modeling a network with moderate traffic volume and a typical signal phasing at the intersection. This will allow for the evaluation of the network's performance under realistic conditions.

4.3 Intersection Modeling

According to the project work plan, the target location is Scott County's CSAH 18/CSAH 21/Southbridge Parkway intersection. Figure 4.1 shows a map view from the area. The current study is going to be demonstrated for northbound CSAH 21. Thus, red light running and warnings are modeled for this direction (however it is simply possible to enable the method for any given direction and route).

The simulator platform allows users to manually draw the road or network and determine properties. Also, it is possible to use available map tools to load any given network. For this study, project intersection is loaded using OpenStreetMap. Figure 4.2 shows how mentioned location is finally modeled as a network scenario. Each direction is increased to five lanes just before the intersection. Two lanes are assigned for the left turn, one lane is for the right turn, and two lanes in middle are determined to go straight. This structure is similar to what is actually available in the field. Comparing map view in Figure 4.1 with modeled intersection in Figure 4.2 represents that coordinations are similar to available maps and documents. Even horizental allignments such as curves are

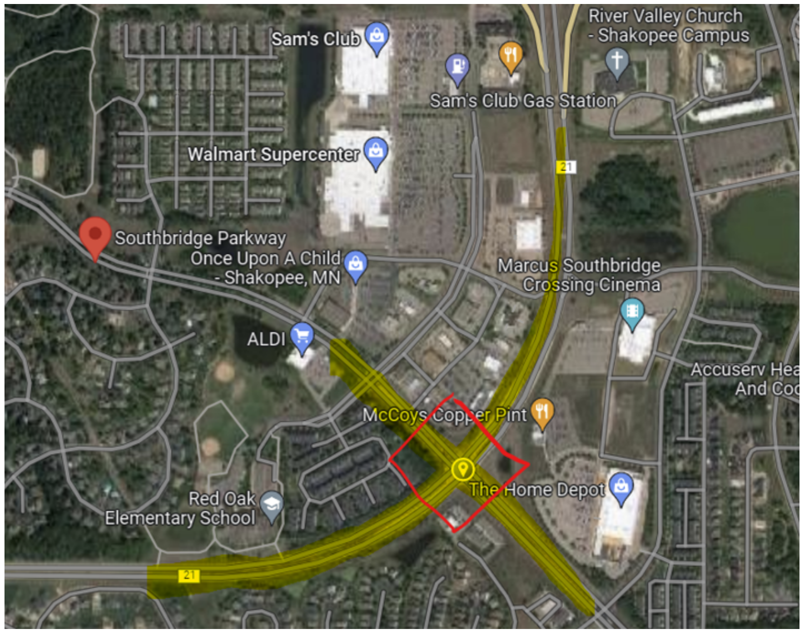


Figure 4.1: project intersection map view, target routes are highlighted

modeled accurately as much as possible.

Notwithstanding the appropriate functioning of the web wizard, errors may occur in certain aspects such as the number of lanes, signal phases, speed limit, etc. To ensure accuracy, the web wizard's output is checked using Google Maps and Google Street View.

After loading and drawing the network, the next thing is to input traffic demand. This is basically how many vehicles are passing through the intersection from different directions. To keep assumptions realistic, the model is loaded with moderate traffic. To make it more obvious when someone runs a red light, scenarios include a single car that runs the red light.

4.4 Methods

The primary objective of this study is to observe and monitor the behavior of drivers who run red lights, in order to develop a suitable model for this phenomenon. To achieve this, it is imperative to comprehend the characteristics of this behavior and to have the ability to accurately identify it. It is evident that drivers who run red lights disregard traffic laws and fail to stop at intersections when required to do so, unlike normal drivers.

The proposed methodology involves first gaining an understanding of this behavior and subsequently devising a pattern to follow it when red light running is detected. Since red light running is not defined as an activity in SUMO, it is only possible to capture this behavior through a dedicated methodology and programming a simulation that can accurately represent it. The project team came up with some methods that are capable to

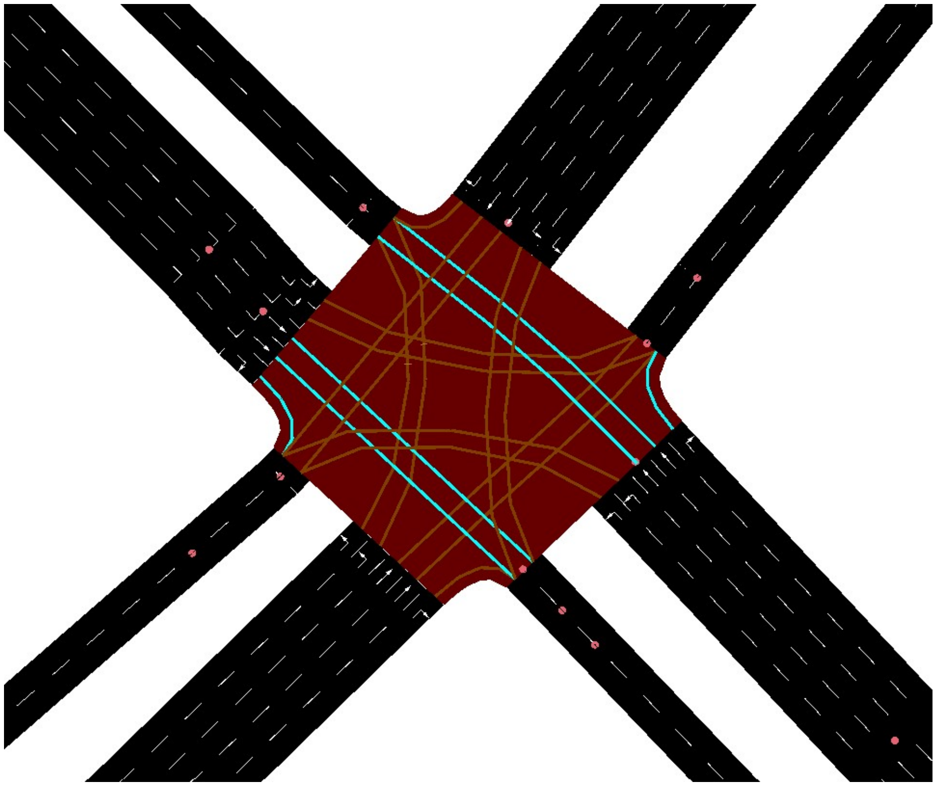


Figure 4.2: Capture of what modeled in SUMO software as the project intersection

program in microsimulation. In this section, three various methods are developed to first follow a red light running and then demonstrate it in SUMO. How realistic they are in addition to their Side effects is evaluated in separate sections.

The primary challenge lies in the fact that standard vehicles in SUMO adhere to traffic rules, such as coming to a stop at red lights. Therefore, the proposed approaches will focus on either modifying the vehicle type or altering network properties to facilitate the demonstration of red light running. Subsequently, this vehicle can be controlled through a displayed warning.

The traffic demand for a given scenario can be defined by the user, who has the ability to specify various types of vehicles that will be present. Each vehicle in the simulation must belong to a specific type, such as "bus" or "car". If the user modifies certain characteristics for a specific vehicle type, those changes will only affect vehicles in that type. Other vehicles within the scenario will remain unchanged.

Using the SUMO platform provides users with the ability to monitor and update vehicle characteristics in real-time. For instance, the speed of a particular vehicle can be set for each time step during the simulation. Most methods for demonstrating red light running and controlling the vehicle will adhere to these criteria. For each method, pros and cons is mentioned separately.

(a) Emergency Vehicle approaching a red light while its blue light is enabled (blue vehicle)

(b) Emergency Vehicle ran a red light (blue vehicle

Figure 4.3: How Emergency Vehicle responds to a red light when its blue device is activated

4.4.1 Emergency Vehicle

Emergency vehicles are granted priority on the road and are permitted to run red lights if necessary. Emergency vehicles are like ambulances and police vehicles. Given their unique privileges, it may be possible to leverage the behavior of emergency vehicles to model red light running in a simulation and see a red light runner acting like an emergency vehicle.

Various types of vehicles can be assigned in traffic simulations, including the "Emergency Vehicle" type, which is defined in the SUMO software. However, the privileges associated with an emergency vehicle will not be activated unless the user explicitly activates the "Blue Light Device" class. The blue light indicates that the vehicle is operating with special rights, such as the use of a siren and blue flashing lights [33]. To model a red light runner vehicle in SUMO, it may be possible to assign it the "Emergency Vehicle" type and activate the blue light device when it approaches an intersection. This will allow the vehicle to ignore red traffic signals.

While red light runners and emergency vehicles may exhibit similar behaviors, there are several side effects associated with converting a regular vehicle to an emergency vehicle. For instance, emergency vehicles are authorized to drive through the opposite direction of traffic or make turns from the wrong lanes. Additionally, emergency vehicles can affect the behavior of other vehicles on the road, such as causing them to move to the side of each lane or stop doing lane changing. These effects are not reflective of red light running behavior and may negatively impact the simulation of the overall traffic stream. [33]

Given these considerations, the project team has decided to reject the emergency vehicle approach as the primary method to model red light running. While the approach is relatively quick and easy to implement, it is not well-suited for accurately reflecting red light running behavior in the simulation. Instead, the team will explore alternative methods that more closely align with the actual behavior of red light runners on the road.

4.4.2 Safety Properties

Drivers have a set of safety considerations in mind when approaching an intersection, such as which direction they need to turn and what actions they need to take to make a safe maneuver. They adjust their speed accordingly and stop when necessary to avoid a potential collision. However, a driver who runs a red light may exhibit looser safety considerations and take the risk of going through the intersection even if it could result in a collision. This method aims to model this type of behavior by modifying safety consideration.

The literature review indicates that most red light violations occur within the first few seconds after the light changes from yellow to red. A study reveals that more than 95% of red light violations happen in first 2 seconds of switching [34]. To model this type of behavior, SUMO offers three safety models: car following, lane-changing, and junction. The junction model is particularly significant for intersections and has parameters that affect safety-related junction behavior, such as red light violation after switching yellow to red. The target parameter in this method is the time after the light changes from yellow to red when a vehicle can still run the red light.

The simulation adds the capability of violating the red light a few seconds after the light switches to red if the algorithm recognizes a particular vehicle. The user can set the value for how many seconds after the light switches to red a vehicle can run the red light. For example, a value of 0 means that drivers can only run the red light during the yellow phase, while a value of 3 means that a red light runner can be seen to run up to three seconds after the light switches to red.

However, this method has a major side effect. All modifications are executed at the intersection, and safety properties are generally related to the network rather than the traffic. Therefore, it is not possible to set a specific junction property for a particular vehicle. Instead, the project team made it possible to demonstrate red light running behavior for any given vehicle by constantly monitoring the intersection. When a vehicle is recognized as violating the red light, the junction property for the intersection is modified to allow the capture of that particular red light runner. During this time, the intersection is modified for all approaching vehicles, not just the offending vehicle. It is challenging to simulate the whole traffic stream accurately during this period considering edited safety properties.

4.4.3 Speed Mode

In order to demonstrate red light running, it is crucial to consider the behavior of a single vehicle. As previously mentioned in the report, drivers constantly adjust their speed to ensure safe driving given the surrounding conditions. SUMO software provides various parameters for vehicles' speed mode, including safe speed, minimum and maximum acceleration, right of way at intersection, and the ability to brake hard to avoid running a red light. To simulate red light running, users can modify these parameters [35]. Disabling some of these parameters including safe speed, maximum and minimum acceleration, allows a vehicle to run a red light. If a vehicle runs a red light, it ignores its safe speed while passing through the intersection.

$$V_i(t) = \begin{cases} 0 & \text{if red light runner} \\ c > 0 & \text{if normal vehicle} \end{cases} \tag{4.1}$$

where $V_i(t)$ is speed of vehicle i in time step t. If a red light violation is appearing, speed at the location of intersection has value of c which is non-zero (it could be chosen depending on the free flow speed and what drivers usually follow in green phase). Figure 4.4 depicts demonstrating a red light running modeled by modified speed mode, while other vehicles stopped behalf of safe speed equal to zero.

Technically, the model is implemented and defined in following steps:

1. First Add another modified speed mode for the case of allowing red light violation
2. Second Every vehicle approaching to the intersection if is in determined range
3. Third Recognize the target vehicle which is trying to run the red light
4. Forth If it is a red light runner, chose the modified speed mode, otherwise use the default speed mode
5. Fifth Check that red light runner will violate the red light according to determined safe speed

It is important to note that the red light runner still follows the car-following model properties, ensuring that there are no concerns with this method. Due to declared reasons, speed mode modification method is selected as the main method to demonstrate red light running in this study. Also speed mode is applied for control strategy when warning issued.

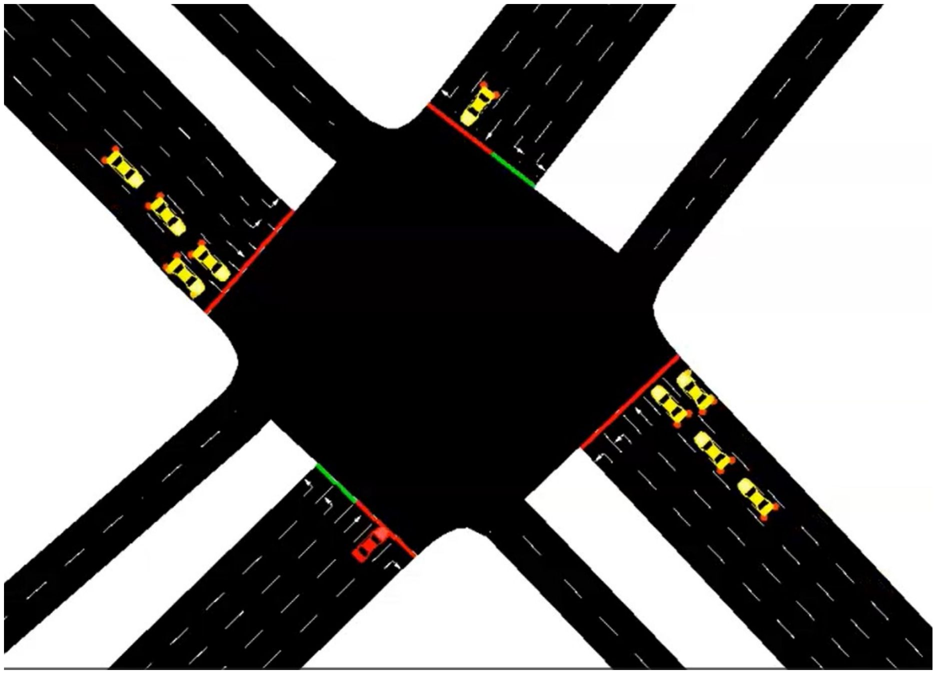


Figure 4.4: Red colored vehicle is recognized as red light running modeled with modified speed mode method

4.5 Warning and Control

Connected technology developed by SPaT leads the control strategy. The connectivity established between the traffic signal and the vehicle enables the proposed method to frequently monitor and update the algorithm. The installed devices provide real-time information about the location and speed of each vehicle, which is then fed into the algorithm to detect potential red light violations. When the algorithm is certain that a red light violation is occurring, a warning is issued to the particular vehicle. With access to the vehicle's speed and location data, the goal is to ensure a safe stop before entering the intersection by computing the required deceleration rate. This can be viewed as a kinematic problem, where the deceleration rate is determined based on the current speed and location of the vehicle.

At the instance of recognizing the red light warning, it is required to provide a stop in a certain distance; the acceleration is calculated in Equation 2. In this equation, it is assumed that deceleration occurs with no delay or reaction time. In other words, if a driver is modeled, reaction time is not included in this equation; it is assumed modeling happens just when pushing the brake. It is possible to add reaction time to the equation but main reason to remove that is emergency brake will be enabled automatically. Therefore this automated mechanism works without delay and will be activated quickly after warning. If the warning deployed, control mechanism would be enabled.

$$V_i(t)^2 = V_0^2 + 2a\Delta x \tag{4.2}$$

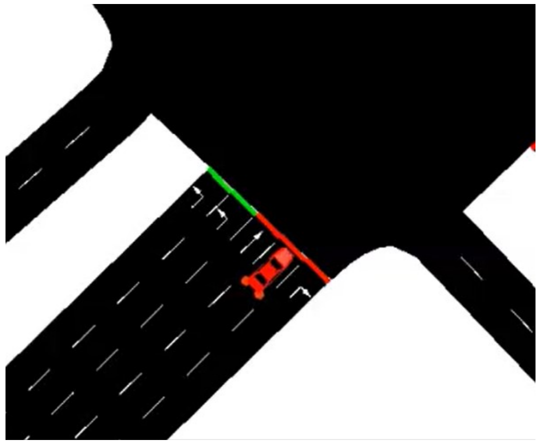


Figure 4.5: After capturing red colored vehicle as red light running, emergency brake is issued and control provided in simulation

where $V_i(t)$ is speed of target vehicle at the moment t (m/s), moment t is just the time that vehicle is recognized as red light runner, V_0 is the speed at the end of control process which is zero (vehicle will be stopped, a is the deceleration rate to be calculated (m/s^2), Δx is the distance from the front bumper of target vehicle to the stop line at the intersection. The deceleration rate is assumed to be constant while stopping is the equation but it is available to model this braking process more rational in SUMO. Deceleration rate could be started from zero, by next time steps increase to reach a peak point, and finally reduce to zero when the vehicle stops.

As mentioned, when mechanism detects a red light running, the deceleration rate is computed. Since the vehicle is violating the signal, emergency braking must be activated, which is a relatively hard brake. This process can effectively prevent the vehicle from running through the intersection, as long as the computed deceleration rate is less than or equal to the maximum available brake deceleration. However, if the vehicle is traveling at a very high speed or the detection is delayed, the target vehicle may stop beyond the stop line.

The described method is implemented in this study. The "Traffic Control Interface" (TraCI) is a tool that provides a way to interact with and send specific commands to simulated objects in the SUMO microsimulation environment [36]. TraCI enables on-time changes to the states and values of the simulated objects, using a programming language. This feature allows the project team to focus on a specific vehicle and modify its dynamics during specified time intervals. The changes made using TraCI are implemented within the simulation during the given time steps and it can be interrupted by user. Therefore Traci can set a determined speed or acceleration rate for a certain vehicle. The program is updated each time step synced with simulation and will set the speed for target vehicle considering the derived deceleration rate.

The proposed method provides the necessary control logic for preventing red light running, and the project team successfully modeled scenarios using this control method. A sample scenario is shown in the Figure 4.5, where the target vehicle was detected as a potential red light runner. If the control had not been implemented, the vehicle would have crossed the stop line. However, with the control enabled, the vehicle is forced to stop by applying the maximum available deceleration rate (4.5 m/s^2 for this scenario). The tail lights of the red vehicle indicate that the control strategy has activated the emergency brake.

4.6 Scenarios

The methods developed in this project are designed to operate under any conditions when a vehicle approaches an equipped intersection. However, to validate and demonstrate the effectiveness of these methods, various traffic demand with different number of vehicles were simulated in scenarios. The update rate for the simulation's time step is set to 0.1 seconds to ensure sufficient resolution. Length unit is meter in SUMO. The length of a regular vehicle is set to 5 meters (16 ft), with a maximum acceleration of 2.6 m/s^2 and a maximum braking deceleration rate of 4.5 m/s^2. For scenarios with emergency vehicle, that certain vehicle had different visualization reflecting its specific rights. A sample signal timing with three phases (green, yellow, and red) is used for the signal with different signal groups. It is obviously possible to adjust phasing and timing for this signalized intersection according to any timing plan.

According to the methodology section of this study, three different methods were proposed. However, since the first two methods were not selected for the final demonstration and control, only one scenario is developed for each. In contrast, the chosen method, speed control, is presented in more detail and applied in multiple scenarios.

4.6.1 Scenario for Emergency Vehicle

In a scenario, two types of vehicles are defined: regular vehicles with normal driving behavior and emergency vehicles. The emergency vehicle is a predefined vehicle type in SUMO, distinguished by its longer length and higher priority, as outlined in the methodology section.

The simulation was implemented on a single lane road leading to an intersection, with a moderate traffic volume of 500 vehicles per hour. Initially, all vehicles enter the road section as regular vehicles, and the algorithm starts monitoring their behavior. If a red light runner is detected, the vehicle's type is immediately changed to an emergency vehicle. The SUMO simulator allows the emergency vehicle to run the red light, as demonstrated in the simulation. Figure 4.3a depicts this process.

4.6.2 Scenario for Safety Properties

A simulation is generated with only regular vehicles on a single lane road, and a moderate traffic volume of 500 vehicles per hour is loaded. In this case, if a red light violation is detected, the safety properties of the intersection are modified in real-time, allowing for a red light running demonstration.

To achieve this, the project team aims to program a command that changes the value of a parameter called "jmDriveAfterRedTime". This parameter controls how many seconds after the signal switches to red, vehicles can still run the red light in the intersection. The default value of this parameter is -1, which means that vehicles only have one second left from the yellow phase before switching to red. To enable red light running, this value is changed to a larger value, but only when the red light runner vehicle is approaching the intersection. Once the vehicle passes the signal, the parameter is returned to its default value.

4.6.3 Scenario for Speed Mode

Since the speed control method was selected to implement for the project study, all scenarios are simulated at the modeled network of the intersection. Additionally, red light running is demonstrated for the target approach of the intersection which has two lanes for going straight, two left turn lanes, and one right turn lane. All vehicles enter the modeled edge of the road with an initial speed of 10 m/s, and the speed limit is set at 25 m/s (55mph). Since the traffic volume is not severe in the scenarios, the vehicles drive at the speed limit when they approached the intersection.

During the simulation, each vehicle follows a certain speed mode. Two speed modes are available for the scenarios. The default speed mode in SUMO requires drivers to follow the rules and stop at the red light. When a vehicle approaches a red light, the speed mode selects a speed of zero for the vehicle. The second speed mode is programmed by the project team, and it allows a value larger than zero for the speed, which allows the vehicle to ignore stopping at the red light. All vehicles enter the road with the default speed mode.

In the first scenario, only one vehicle approaches the intersection when the traffic signal is in the red phase for that direction. This vehicle does not provide appropriate deceleration before the stop line, and the algorithm detects it as a red light violation. Therefore, the speed mode for that vehicle immediately changes from the default to the second speed mode, which allows the vehicle to ignore the red light at an empty intersection.

The second scenario involves a regular vehicle passing through a yellow light, which is not considered a violation. However, when a second vehicle follows closely behind, the traffic signal changes to red. Despite this, the second vehicle does not stop and continues forward, resulting in a red light violation. The speed mode for the first vehicle remains in the default mode, while the second vehicle switches to the second speed mode, which enables it to run the red light. Figure 4.6 depicts this scenario for the empty intersection.

To ensure the realism of the scenarios, moderate traffic loads is included in the third and fourth scenarios. A traffic volume of 200 vehicles per hour is loaded for each approach except for the target approach, where the red light violation occurs. The right and left turn lanes have a traffic volume of 60 vehicles per hour. All vehicles in other approaches are programmed to follow the default speed mode, which makes them stop in accordance with

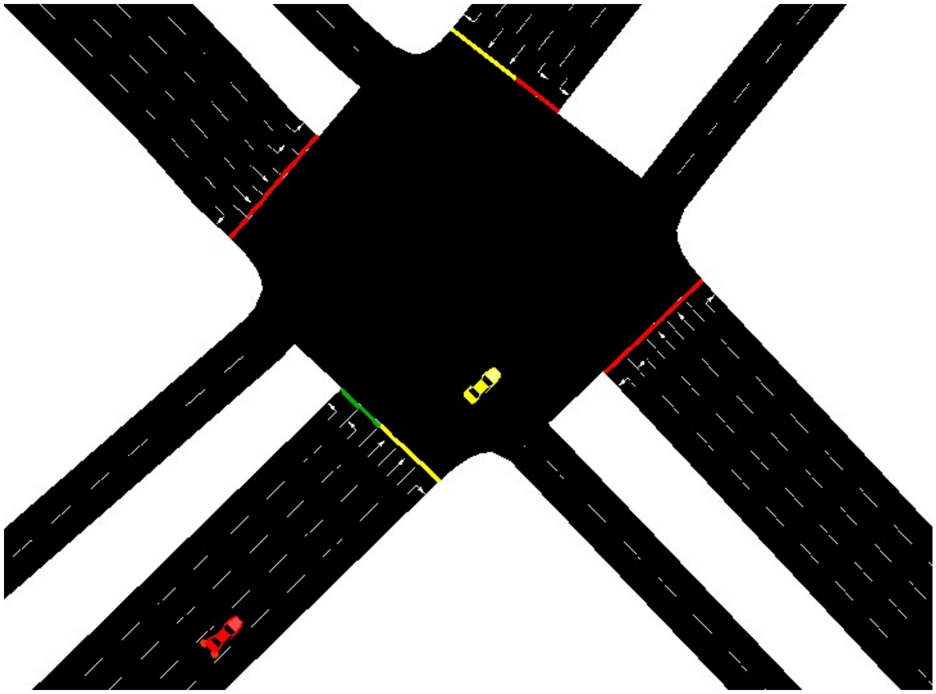


Figure 4.6: Yellow colored vehicle runs yellow light but red colored vehicle will face a red light

the traffic signal phase.

The third scenario involves the detection of a single vehicle violating the red light in the target approach, while all other approaches have moderate traffic demand. Figure 4.7 depicts how the red colored vehicle runs the red light while other vehicles have stopped before entering the intersection. The simulation process is similar to scenario one, with a focus on the single vehicle violating the red light. The fourth scenario is generated in a moderate traffic demand environment and is similar to scenario two, with the only difference being the moderate traffic demand.

4.7 Summary

This task aimed to achieve two primary objectives: demonstrating red light running behavior and providing a control logic in microsimulation. Technical mechanism for identifying red light running will be developed in future tasks using connected technology. Therefore task 3 assumed the mechanism detected red light running among traffic flow. Based on that, various methods were developed to model this phenomenon. Microsimulation was programmed to monitor intersections and approaching vehicles. If red light running was detected, the methods could demonstrate it in microsimulation. Among the proposed methods, modifying speed selection was selected because it had no side effects during simulation.

The application of this mechanism enabled the control of a target vehicle and forced it to stop in microsim-

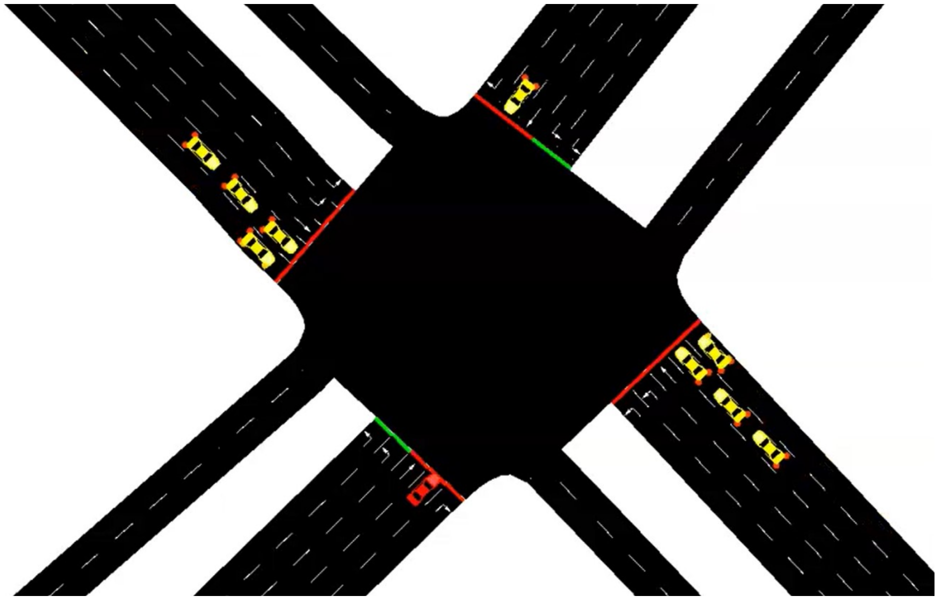
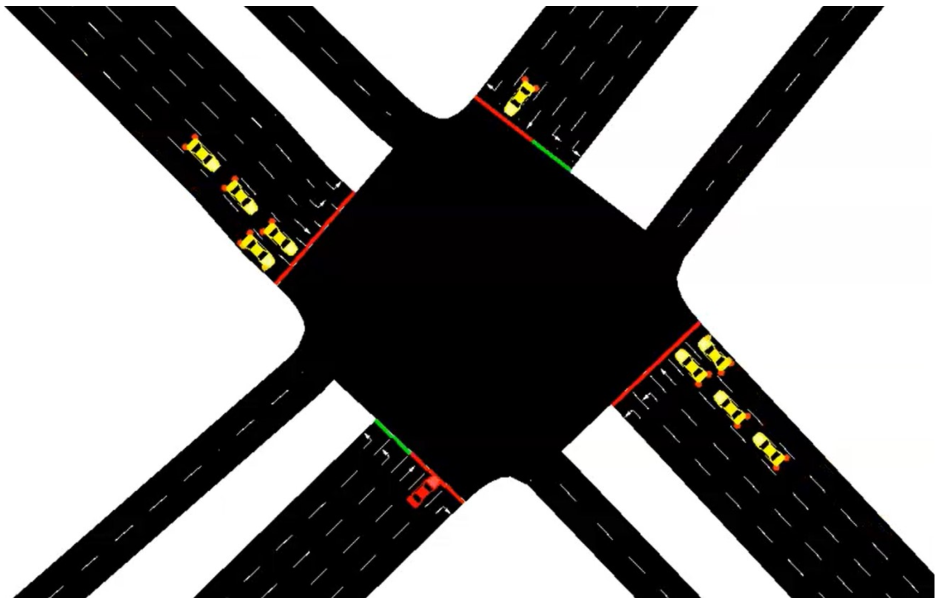


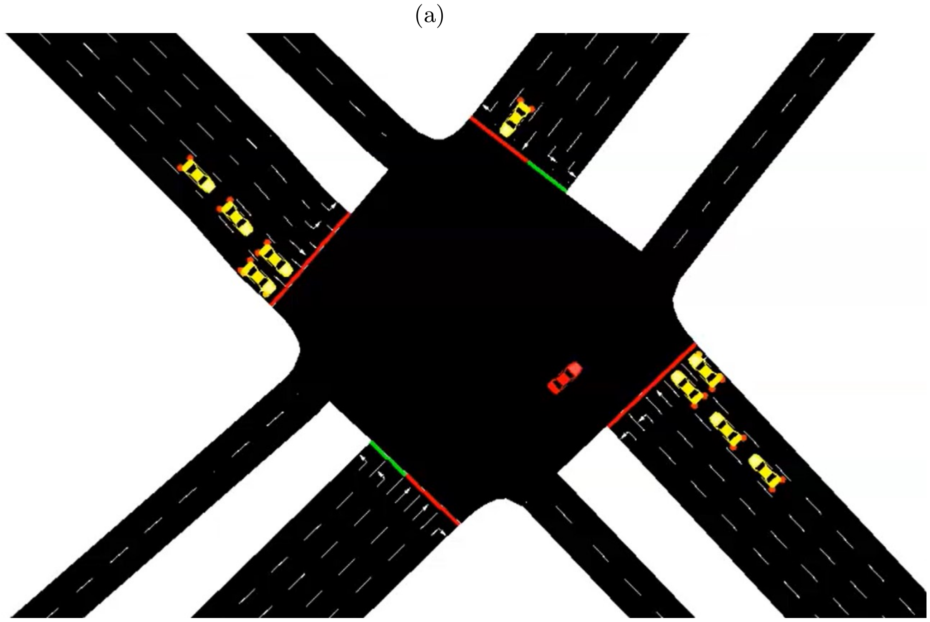
Figure 4.7: A vehicle violates the red light in moderate traffic

ulation. The control logic involved monitoring each individual vehicle approaching the intersection to capture a potential violation. The deceleration value was then determined, and a forced stop was initiated before the vehicle runs through the intersection.

In this task, demonstration and control platform was created, which will be used as a basis for future tasks. The next tasks will focus on developing a detection algorithm that can identify red light runner vehicles. When a vehicle is detected by the algorithm, the platform created in this task will be activated to simulate a red light violation, without any negative impact. Therefore, Task 3 will operate within the overall methodology established for the upcoming tasks.



(a)



(b)

Figure 4.8: A vehicle violates the red light in moderate traffic

Chapter 5

RLRWS for individual target vehicle

September, 2023

5.1 Introduction

Connectivity has introduced a fresh perspective to traffic engineering, specifically through two key aspects: vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. This development has opened up a realm of possibilities, with the potential to enhance road safety, optimize energy usage, and improve overall traffic flow [37] and [38].

In our previous tasks, we have detailed the installation of essential equipment within both vehicles and at signalized intersections. The primary objective was to establish seamless communication between these critical components on the road, with a particular focus on receiving signal timing plans as vehicles approach intersections.

In Task 3, the Red Light Running Warning System (RLRWS) was demonstrated along with gaining control over a target vehicle within a microsimulation environment. In this task, the project team took the reins in developing a red light running warning algorithm. The algorithm was designed to provide advanced warnings to vehicles approaching signalized intersections, thus mitigating the risk of red light violations. The algorithm's development hinges on acquiring crucial data: Signal Phasing and Timing (SPaT) messages and the longitudinal distance between the target vehicle and the signal's stop bar. Leveraging this data, the control algorithm will generate a warning system tailored to each specific scenario.

The demonstration of the RLRWS process begins with receiving the necessary information through measurements and V2I connections. This information, which includes vehicle dynamics and signal phasing plans, is continually updated as the vehicle approaches the intersection. Vehicle dynamics generally encompass two aspects: First, information about other vehicles, especially the preceding vehicle, such as speed and spacing to the target vehicle. This information is transmitted via vehicle-to-vehicle (V2V) connections. Second, the dynamics of the current vehicle, which includes its current speed and coordinates. When implementing the system for an individual vehicle, there is no need to anticipate the motion of the preceding vehicle. The demonstration of the RLRWS in a platoon of vehicles will be addressed in Task 5.

The updated states are then processed through the warning algorithm, which is comprehensively explained in this report. The algorithm calculates the appropriate warning message, namely the corresponding acceleration-deceleration values for a specific horizon. Furthermore, the warning algorithm is designed to be adaptive which means displaying warning messages based on the current vehicle dynamics.

Figure 5.1 explains the aforementioned process. Measurement and connectivity obtain the required data and then update the algorithm which computes an adaptive warning message. The algorithm returns the results to the vehicle to be displayed and control the target CAV accordingly. The process is been repeated as a cycle for the next time step when updated information are available.

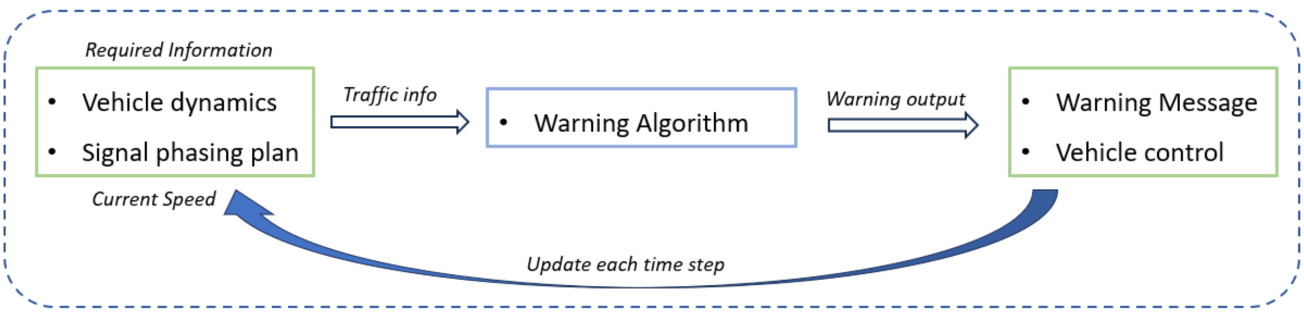


Figure 5.1: An overview of the RLRWS framework

It is crucial to test the proposed RLRWS before implementing it in vehicles and deploying it on the road. Consequently, the RLRWS is modeled and tested in microsimulation to ensure the correctness and reliability of the algorithm's performance. The Simulation of Urban MObility (SUMO) is utilized as the microsimulation tool to control a target vehicle by adhering to the warning messages generated by the algorithm. SUMO simulation is introduced in Task 3 in detail.

The described process is implemented within the SUMO simulation to validate the proposed algorithm. A target vehicle is simulated as it approaches a signalized intersection. The simulation continually updates the target vehicle with the latest vehicle dynamics and signal phasing plans. These updated states are then inputted into the warning algorithm. Ultimately, the computed warning messages are utilized to control the behavior of the target vehicles within the simulation environment.

The warning output will be clear and user-friendly, consisting of a message that guides the driver on the appropriate action to enhance safety. Our ultimate goal is to make intersections safer through this innovative technology by guiding the driver when required. The structure of the current task begins with describing the warning message and control plan, subsequently the required information for the algorithm is prepared, followed by elaborating the warning algorithm. Furthermore, implementing the algorithm into microsimulation is presented. Finally, multiple scenarios of a single vehicle when approaching the intersection are elaborated in order to ensure the performance of the warning system and represent the warning message accordingly.

5.2 Warning Message

The warning algorithm forms an integral part of this system, addressing a complex mathematical problem within the confines of real-world physical constraints. It seeks to derive optimal acceleration values that should guide the vehicle's behavior. However, it is essential to underscore that the target vehicles in this context are controlled by the drivers. The driver retains control of the vehicle at all times and the RLRWS displays messages to guide the driver for a safe stop.

These findings have led the project team to devise a warning message system that can effectively communicate the level of braking required for the impending stop. First, the RLRWS computes the acceleration-deceleration rate during the prediction horizon based on the current vehicle dynamics to stop at the signal location. Second, the warning message is generated based on the algorithm's output, which is then displayed to the driver.

Within this framework, the warning messages are displayed in a colored bar where values represent the accelerate-decelerate intensity with respect to the maximum braking and maximum acceleration rate. Since the project is focused solely on studying the stopping process, the main focus is on deceleration and the acceleration process is studied as normal driving behavior (the driver has regular actions and no guidance is required). Therefore, the color mainly describes the suggested brake intensity. Colors are categorized into three distinct colors: Green, Yellow, and Red, as outlined in Table 5.1. Furthermore, Figure 5.2 displays the full range of all possible warning messages with the corresponding color. The vertical axis is the intensity with respect to the maximum rate. For example, if the maximum deceleration is defined to be 5 meters per second squared, a warning message of 50% means to brake with half power or a rate of 2.5 meters per second squared. This example also will be displayed in the yellow light.

Table 5.1 and Figure 5.2 explain the warning bar where the colors represent the intensity range. the green color represents normal driving behavior, and the RLRWS does not provide any additional guidance, encompassing all positive acceleration values as well as minor acceleration values. The yellow color indicates that the driver needs to make a moderate deceleration, including moderate deceleration rates (or moderate brake intensity rates), while the color switches to red when there is an imminent risk of a red-light violation with the necessity of an intense brake. Generally, the red color is displayed when the driver has not followed the yellow color bar after a few seconds ago. As a result, yellow switches to red, and the brake intensity is increased in the displayed bar gradually as the driver did not follow the message.

Table 5.1: Warning message color classification

Warning Message	**Definition**
Green	No action required, far from signal or signal is green when the vehicle arrives
Yellow	Driver needs to decelerate gradually, a full stop is required soon
Red	A hard brake is required, otherwise will run the red light

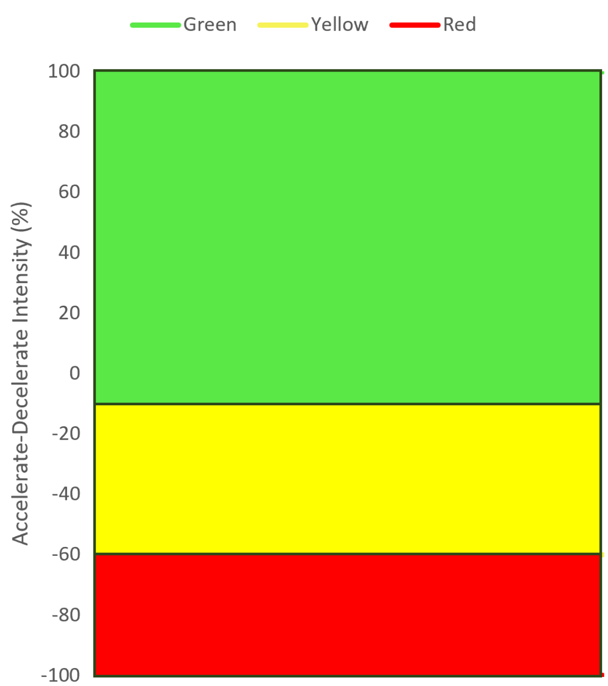
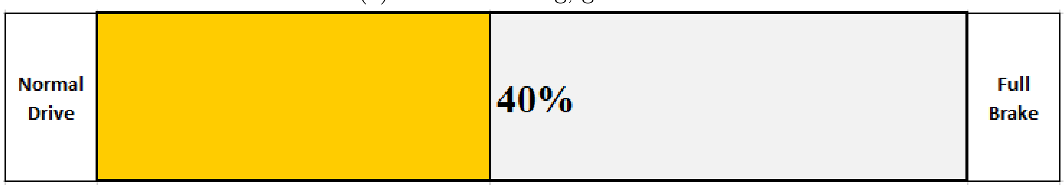


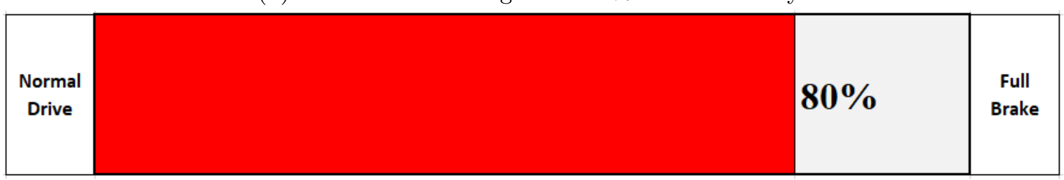
Figure 5.2: Warning message range, where the vertical axis represents the intensity of acceleration when above zero and deceleration when below zero

Figure 5.2 represents a range for all possible messages involving acceleration or deceleration. However, it is important to note that what the RLRWS actually displays to the driver is a warning message based on the algorithm in real-time. As aforementioned, the RLRWS focuses on the braking process, not acceleration. Therefore, the brake intensity is only displayed to the driver, not the acceleration intensity. In the case of acceleration, the system only displays "normal driving" with a green color without specifying the intensity amount. Figure 5.3 illustrates three examples of what is displayed to the driver at each instance. The warning message is provided in a horizontal bar ranging from normal driving behavior to the maximum brake that displays the suggested brake intensity at the current time. Figure 5.3a represents the case of a green color. The text ”normal driving” is only colored green, and the acceleration intensity is not mentioned, which means braking is not required. Figures 5.3b and 5.3c also represent brake intensities with the corresponding value that comes from the algorithm. Since the warning message for these cases ranges between 0 and 100 percent, the bar spans between normal driving behavior and the maximum brake.

(a) Normal Driving, green color



(b) Yellow color message with 40% brake intensity



(c) Red color message with 80% brake intensity

Figure 5.3: Examples of warning message displayed to the driver while driving

Crucially, it should be emphasized that the warning message system operates on a time-dependent continuum. It begins with a Green color, signifying that all conditions are safe with regard to red-light violations. Subsequently, it transitions to Yellow, indicating the need for caution, and finally to Red if the driver fails to initiate sufficient deceleration for a safe stop. Thus, it never immediately starts by displaying a harsh brake intensity via red color. Messages begin with moderate braking intensity with green and yellow color. If the driver does not follow the message which means a red light violation is more possible, the bar intensity will increase to suggest a harsh brake. However, in typical driving scenarios where drivers maintain a deceleration intensity close to the suggested rate, the warning message will not escalate to the Red level. A comprehensive exploration of the warning message display logic is detailed in Section 5.6.

The warning message is updated based on the current dynamics of the vehicle. The warning algorithm in the following sections relies on the initial states, which are the current vehicle dynamics. It is essential to utilize the current vehicle dynamics to maintain an updated acceleration-deceleration strategy. For the red light warning, the latest location to enable the brake is computed using kinematics and is a function of the current speed and maximum deceleration rate. The earliest time for displaying the warning should be when there is enough spacing left to reach the signal. Presenting an intensity ratio scaled to 100% for the warning message is chosen to avoid confusion and simplicity. Most drivers are not familiar with the deceleration value and they do not know how to make a brake that is equal to a given deceleration amount. Instead, drivers are used to their vehicle braking capabilities and what the maximum braking feels like. Based on that, they can press the brake pedal with the desired intensity such as 50% of maximum braking.

5.3 Signal and vehicle dynamics

As mentioned, to deliver warning messages effectively, the algorithm relies on both the current status of the target vehicle and the signal's phasing and timing plan. The vehicle's present conditions are necessary to calculate the appropriate location and severity of the required deceleration. Furthermore, the signal's phasing and timing data informs the target vehicle about the current signal phase, the remaining duration until the next phase, and what the upcoming phase will be.

While the target vehicle approaches the intersection, the algorithm continuously monitors the collected data to make decisions. When the vehicle is a considerable distance away from the signal, it should continue operating under regular traffic conditions. In such cases, the algorithm remains passive and does not intervene in controlling the vehicle. However, as the target vehicle gets closer to the intersection, the algorithm takes into account both the signal's timing plan and the vehicle's current dynamics. It predicts when the vehicle will arrive at the intersection and uses the signal's timing information to assess the signal phase at that moment. If the signal phase indicates that the light is green, the algorithm allows the vehicle to proceed through the intersection without intervention. Conversely, if the signal phase indicates a red light, the algorithm initiates a warning process, ensuring that the vehicle comes to a complete stop before reaching the stop bar.

First, the V2I connection and SPaT messages are assumed to be established by previous tasks. Therefore the current phase, future phases, and corresponding timings are also available. Most significantly, the current phase and duration left of the current phase are required by the model.

Second, the longitudinal distance to the signal is required for the algorithm. OBU devices which are installed on the vehicles, have the latitude and longitude coordinates constantly. In addition, V2I connectivity provides the actual signal coordinates.

Distance to Signal

The distance between the location of the current vehicle and the signal stop bar is not the difference between the latitude and longitude coordinates of those two points. Vehicles are traveling toward a road with complicated topography not a straight line. Therefore, we calculate the true distance based on the distance traveled by monitoring the path where the previous vehicle has completely crossed through the intersection.

For the target intersection in the project when implementing the RLRWS in reality, precise coordinates and profiles of all incoming and outgoing roads (streets, arterial roads, etc.) within the connectivity range (the maximum distance that CVs can receive SPaT messages) are predefined and loaded. These values are stored as map messages for the signal control unit. This information can be transmitted to the target vehicle via V2I. The data received by the target vehicle serves as a reference list and the following process can be applied to the actual roads. In this task and for the modeling stage, a reference vehicle is simulated only once for the target approach, and the x and y coordinates are recorded during this process for each simulation time step.

Each x and y represents a point in the two-dimensional space. The recorded points generate a reference list for the following sections. The distances between each of two adjacent points are computed by Equation (5.1) for entire elements in the reference list. These distances represent the traveled distance between those two points:

$$d_{i,i+1} = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{5.1}$$

where i is the index of an element in the reference list.

The location of the signal is also available from V2I. Among the points in the list, the closest point to the signal location is computed. This point represents the closest location that the reference vehicle had to the signal and signifies the signal location. In the reference list, all points after that will be ahead of the signal, while earlier points are before arriving at the signal. Traveled distances are used to compute cumulative traveled distance starting from the element of the signal location, going backward and forward separately. Moving backward from the signal location explains the distance left to the signal, and moving forward explains how much has been passed from the signal.

The aforementioned process computes the distance remaining to the signal in each location. In order to develop the algorithm and be able to implement it for a target vehicle, it is required to update the remaining distance for each time interval. Therefore, the current x,y coordinates of the target vehicle are compared to the reference list. We try to find the closest point in the reference list to the current location of the target vehicle called J_c in Equation (5.2).

$$J_c = \arg\min_{i=1}^{n} \left[(x_i - x_t)^2 + (y_i - y_t)^2\right] \quad \text{for every } i = 1 \text{ to } n \tag{5.2}$$

where n is the number of points in the reference list, x_t and y_t are the current coordinates of the target vehicle.

The final step is to find the remaining distance from that element in the list to the signal location. Since we already computed the cumulative distances for each element, the cumulative value is approximately the target distance. It should be noted that if the algorithm time step is chosen to be large, major errors will be added to the values particularly when the vehicle is close to the signal. In order to address this issue, this study is using a time step of 0.1 second for simulations.

5.4 Warning algorithm

In this section, we show the formulation of the proposed red light running warning system. The proposed algorithm will be formulated in the model predictive control (MPC) fashion, which calculates the optimal input of a control system under several constraints given the initial condition. The MPC algorithm can optimize the system's input based on the system's predicted future states. This is suitable for the development of the RLRWS, which requires the algorithm to consider the future behaviour of a driver given the future signal status. Fig. 5.4 shows the general framework of the proposed system. The input of the system is the traffic condition, which includes the target vehicle's speed and location information, signal phase and timing (SPaT) information from traffic signals that are in the V2I range. The output of the system is the warning signal, which suggests the target vehicle's driver to do reaction in advance to avoid running a red light. Two modules are included in the system: traffic prediction algorithm and MPC-based optimization problem. The proposed system will update the warning signal every 1 second and details about the system will be discussed in the following subsections.

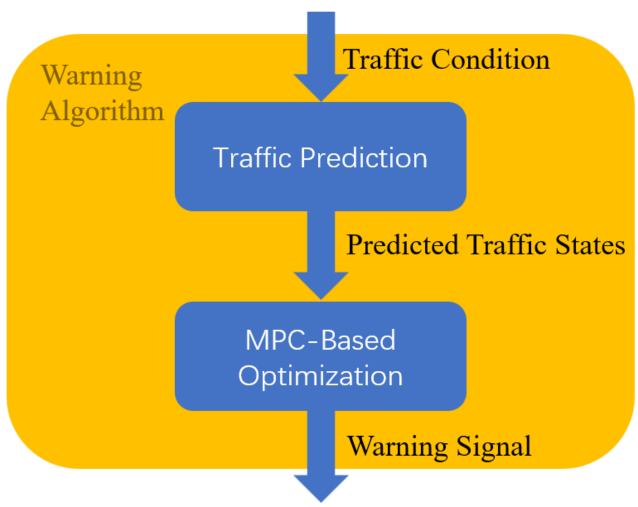


Figure 5.4: A framework of proposed red light running warning algorithm. The input of the system is the traffic condition and the output of the system is the warning signal to the vehicle's driver.

To provide the warning signal to the vehicle's driver in advance, the algorithm should have the capability to predict future traffic states. The traffic prediction framework proposed in [39] is used in the proposed system. Using the target vehicle's speed and location information, along with the SPaT information from nearby signalized intersections, the traffic prediction algorithm will predict the target vehicle's longitudinal trajectory for the next 15 seconds. This predicted trajectory will be used by the MPC-based optimization problem. The following formulation will be used:

$$u^*(t)=\text{argmin}\int_{t_0}^{t_f} q(x(t), v(t), a(t))dt \tag{5.3a}$$

$$\text{s.t.} \quad \dot{x}(t) = v(t), \tag{5.3b}$$

$$\dot{v}(t) = a(t), \tag{5.3c}$$

$$a(t) = f(u(t), x(t), v(t)), \tag{5.3d}$$

$$0 \leq u(t) \leq 100 \tag{5.3e}$$

$$v_{\min} \leq v(t) \leq v_{\max}, a_{\min} \leq a(t) \leq a_{\max}, \tag{5.3f}$$

$$x(t) \leq x_{\text{tl}} - v(t)\tau_0, \text{if the traffic signal is red at } t, \tag{5.3g}$$

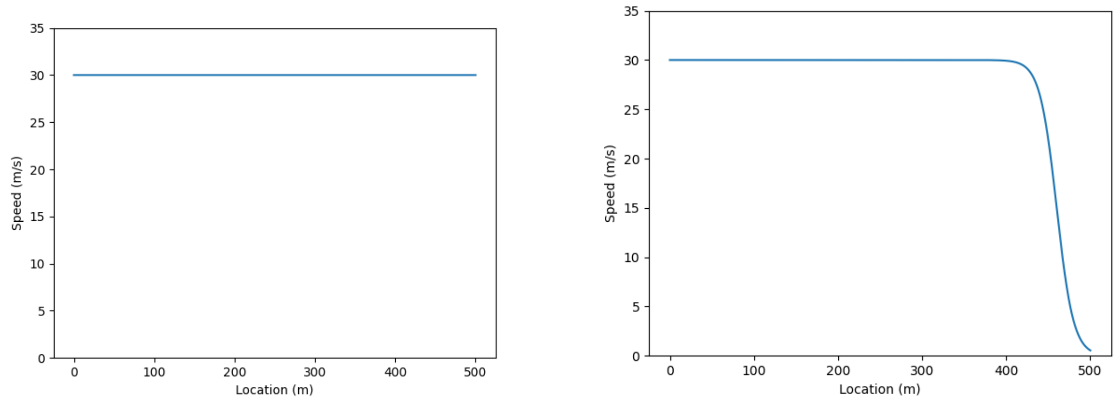
$$v(t_f) = 0 \text{ and } x(t_f) \geq x_{\text{tl}} - d_0, \text{if the traffic signal is red at } t_f \text{ and } x_{\text{pred}}(t_f) \geq x_{\text{tl}} - d_0 \tag{5.3h}$$

$$x(t_0) = x_{t_0}, v(t_0) = v_{t_0} \tag{5.3i}$$

where $u(t)$ is the warning signal provided to the driver; $x(t)$, $v(t)$ and $a(t)$ are the vehicle's longitudinal position, longitudinal speed and longitudinal acceleration, respectively; t_0 and t_f represent the beginning and end time step of the optimization problem; x_{tl} is the longitudinal position of the traffic signal; τ_0 represents the desired time headway; d_0 is a small distance value; $x_{\text{pred}}(t_f)$ indicates the vehicle's predicted longitudinal location at the end time step of the optimization problem; (5.3a) shows the optimization's objective function; function $f(\cdot)$ is the driver model, which shows the driver's reaction to the warning signal; (5.3e) represents the constraint on warning signal's value; (5.3f) indicates the physical constraint on vehicle's speed and acceleration; (5.3g) and (5.3h) are used to avoid running a red light and will be explained later; (5.3i) is the initial condition of the optimization problem. This optimization problem could be solved using any type of nonlinear numerical solver.

The objective function is employed to describe the cost of vehicle's movement along the prediction horizon. In this task, the objective function is set to $q(x(t), v(t), a(t)) = w_1 a^2(t) + w_2 \dot{a}^2(t) + w_3(v(t) - v_{\text{ref}})^2$, where w_1, w_2 and w_3 are weighting factors; v_{ref} is a reference speed as shown in Fig. 5.5. In the figure, we assume the traffic signal is at the 500 m. When the traffic signal is green, the reference speed is the free flow speed of the road. When the traffic signal is red, the reference speed drops before the stop bar. To simplify the computational process of the solver, this reference speed is modeled in the form of a sigmoid function. As a result, the proposed objective function will minimize the vehicle's acceleration and jerk while tracking the reference speed along the prediction horizon, which provides a smoother longitudinal maneuver.

Given a warning signal, the driver will react to it by adjusting the vehicle's longitudinal speed $v(t)$. However, when the vehicle is driving at different speed or is locating at different longitudinal location, the driver's reaction to the



(a) Reference speed when the traffic signal is green. (b) Reference speed when the traffic red is red

Figure 5.5: Reference speed used for MPC optimization.

same warning signal will be different. For example, when the vehicle is far away from a signalized intersection, the driver tends to apply a smaller force to the braking pedal compared to the scenario when the vehicle is close to the red light. Therefore, the real relationship between the warning signal and the vehicle's acceleration also depends on the vehicle's location and speed. In this work, the function $f(\cdot)$ is used to describe this. For Task 4, a linear driver model $a(t) = -u(t)/20$ with $0 \leq u(t) \leq 100$ is used as this driver model. It is obvious that drivers may not follow this model perfectly and will have different behaviours. However, since the MPC problem will update the warning signal every 1 second, the algorithm will use the latest vehicle state at each update instance as the initial condition. For example, a larger warning signal will occur if the driver doesn't decelerate the vehicle at the beginning.

At the beginning of each optimization circle, the prediction algorithm will predict the vehicle's longitudinal trajectory along the prediction horizon. If the vehicle is predicted to reach the stop bar at the end of the prediction horizon t_f under red light (reach a certain range ahead of the stop bar, e.g. the vehicle is predicted to be within 20 m ahead of the stop bar at the terminal horizon), the constraint (5.3h) will be included in the optimization problem. This will force the vehicle stop close to the stop bar at the time t_f. To avoid infeasibility of the problem, positive slack variables γ_v and γ_x are included in the optimization problem and constraint (5.3h) is re-written as: $v(t_f) == \gamma_v$ and $x(t_f) + \gamma_x \geq x_{\text{tl}} - d_0$. Term $w_v\gamma_v^2 + w_x\gamma_x^2$ with two large positive weighting factors w_v and w_x is added to the objective function (5.3a). In this way, the optimization problem allows the problem to slightly violate these constraints when they cannot be satisfied, which guarantee the feasibility of the algorithm.

When the ego vehicle is approaching the stop bar and the traffic signal is red, shorter prediction horizon is required by the MPC optimization and the terminal constraint (5.3h) on distance will be re-written in a stricter form. This is because as the vehicle is approaching the intersection, it requires shorter time to reach the stop bar and the vehicle is expected to stop closer to it. In this problem, the initial prediction horizon is 15 sec and d_0 is set to 20

m. When the ego vehicle is 60 m away from the stop bar, these two values will be updated to 10 sec and 15 m; When the ego vehicle is 40 m away from the stop bar, these two values will be updated to 8 sec and 10 m; When the ego vehicle is 20 m away from the stop bar, these two values will be updated to 6 sec and 5 m. This process will speed up the computational process and force the vehicle to stop before the stop bar completely.

5.5 Simulation implementation

As explained in previous tasks, SUMO simulation is employed to demonstrate the algorithm for a target vehicle. In this task, the V2I communication range is assumed to be 500 m. There are many simulations for modeling traffic behavior but most of the simulation platforms follow a static network loading and are not programmable as well. In other words, the traffic volume should be repetitive, and changes on a microscopic scale such as focusing on a certain vehicle are not possible. However, SUMO includes an API that allows the user to interact with the simulator, adjust vehicle behavior, and control any vehicle by developing a desired algorithm. Therefore, programming and live interaction are available in SUMO which is why we choose to implement the RLRWS in SUMO. The project network including the intersection zone is defined in previous tasks. The current task is focused on a single vehicle RLRWS. Therefore, the traffic demand consists of a single passenger vehicle which follows the maximum speed limit or the road. Table 5.2 explains the defined parameters for the vehicle physics.

Additionally, SUMO is capable of retrieving the current vehicle speed, assuming it originates from sensor measurements or vehicle onboard units. The computation of acceleration, coupled with the current speed, leads to the determination of the speed for the next time step, facilitating the control of the target CAV.

Table 5.2: Loaded vehicle characteristics into SUMO

Parameter	**Value**
Length (m)	5
max speed (m/s)	30 (67 mph)
max acceleration (m/s^2)	2.6
max deceleration (m/s^2)	5
max deceleration (m/s^2)	5

The simulation starts running the defined network by loading the defined vehicles with respect to time steps. As mentioned in Figure 5.1, the simulation time step is chosen to be 0.1 seconds in order to ensure accuracy. In each time step, the vehicle coordinates are compared to the reference list which results in the remaining distance to the signal. Short time intervals guarantee to find the closest coordinate to the actual coordinates. Furthermore, using short time intervals for the SUMO section will not result in high computation time since there is minor computation in the SUMO section. The main computation is within the warning algorithm which will be updated every 1 second rather than each time step to handle the computation time.

The warning algorithm will compute a list of warning messages for the warning algorithm prediction horizon (forward-looking time interval that the model predicts the information) which is the next 10 seconds. This list is split into 0.1-second time steps to demonstrate the control. As discussed in previous sections, the warning message is a continuous brake intensity with a color as driver guidance which is modeled in SUMO accordingly. In each second, the algorithm is run based on the current vehicle states which results in an acceleration list. Values for the next second which includes 10 time steps, are employed to set the vehicle speed for the next 10 time steps. At the beginning of the next time step, the algorithm will be updated again. Therefore, the only values corresponding to the next time step will be used for the speed control. Furthermore, the speed of the next time step is computed as:

$$v(i+1) = v(i) + u(i) \times dt \tag{5.4}$$

where $v(i)$ is vehicle speed at time i, $u(i)$ is the warning signal value computed by algorithm from Equation (5.3a) and i is the current time step. It assumed that the driver followed the warning message provided for the next time step with a relatively small error.

It is assumed that the maximum acceleration rate for the vehicle is 3 meters per second squared (9.84 feet per second squared) and maximum deceleration is 5 meters per second squared (16.40 feet per second squared) for the simulation process. The maximum rates are adjusted based on the vehicle type and the warning message will be displayed based on them. There are also assumptions regarding the warning message and the corresponding color. When the vehicle is within the connection range and the acceleration rate computed by the warning algorithm is above zero or smaller than 0.5 meters per second squared (10% brake intensity in the warning message) for the deceleration rate, the warning message has a green color. When the rate exceeds 0.5 meters per second squared of deceleration (10% brake intensity), the warning message switches the yellow light.

If the driver follows the warning message, the algorithm will remain in the generated plan and corresponding brake intensities for the prediction horizon. However, if the message is displayed but the driver does not take any corresponding action, the message will be updated for the upcoming time steps based on the current driver speed not the predicted speed value by the warning algorithm. Therefore, as the vehicle approaches the intersection, the brake intensity rates become higher. It will switch to the red color and a hard brake suggestion if the driver does not take any action until the last seconds. The scenarios for these described cases will be assessed in the following section.

Furthermore, if the warning message corresponds to higher intensities and then switches to lower values as it gets closer to the intersection, even if the intensity falls below 10% (the threshold for switching from yellow to green), the warning message will continue to be displayed with the yellow and red colors. The reason for not switching to the green color is that the driver might become confused and attempt to accelerate while the signal is still red. When the vehicle is close to the intersection and the signal turns green, the algorithm obtains the required acceleration but since the focus is on the brake process, the warning message only displays a green color which means normal driving behavior. In this case, the system does not suggest any deceleration or acceleration intensity to avoid showing additional information to the driver.

5.6 Evaluation for scenarios

This task aims to propose a warning algorithm that provides optimal deceleration rates and ensures a complete stop before the stop bar and for a single vehicle. Therefore, this section primarily assesses the performance of the proposed algorithm for all possible real-world scenarios. The signal phase when the vehicle is approaching the intersection and when it arrives at the signal location is a key parameter, in addition to whether the driver follows the warning message consistently or not. Therefore, the modeled scenarios include the green phase and red phase. The yellow light is considered an extension of the green light since vehicles are still permitted to proceed through the intersection. There are two possible cases for each signal phase. First, the signal phase is consistent when the vehicle is approaching it. The duration of consistency means only when arriving at the signal is within the prediction horizon. Otherwise, the vehicle is not aware of the traffic signal. Second, the signal phase is in one color when the vehicle is approaching, and it switches to another phase with a different color. In this case, the vehicle is also aware of the upcoming phase via V2I, and it should also be reflected in the RLRWS message. Therefore, all possible cases in respect to the signal will be as follows:

1. Consistent green light when the vehicle is approaching.
2. Red light at first when the vehicle is approaching and switch to the green light when the vehicle is close to the intersection.
3. Consistent red light when the vehicle is approaching.
4. Green light at first when the vehicle is approaching and switch to the red light when the vehicle is close to the intersection.

In case 2, the signal is green, meaning that the vehicle can freely pass the intersection. Therefore, there is no possibility for the red light warning, and in the scope of the RLRWS study, it can be viewed as a regular green light. Therefore, cases 1 and 2 are described as a single scenario.

Furthermore, concerning the driver's adherence to the warning message, there are various possibilities. The driver could follow the warning message from the beginning, or not follow the messages at first and start following when close to the intersection. Other situations, such as following the message at first and then ceasing to follow it, or following haphazardly, are also possible. However, presenting one scenario of partially following the warning message will demonstrate that the proposed framework can address all other situations as well.

In summary, three cases of changes in the signal phasing are presented when the driver is following the RLRWS suggestions. In addition, a scenario where the driver does not follow the messages at first but will follow when the vehicle is close to the signal is presented in the following sub-sections.

Vehicle dynamics could also be a variable, but driving with the maximum speed and slowing down to zero will cover the speed values in between. For example, when a driver is driving slower than the maximum speed, the RLRWS will suggest deceleration later than when driving with the maximum speed.

In the following scenarios, three plots are presented including a space-time plot, a speed plot, and a warning message plot as the vehicle is approaching the intersection up to when the vehicle passes the intersection. The current status of the traffic signal is depicted with the dashed line in all plots and the location of the traffic signal is marked with a yellow arrow in all space-time plots. As described, the warning message is representative of the acceleration-deceleration values. Therefore, the sole acceleration plot is not presented.

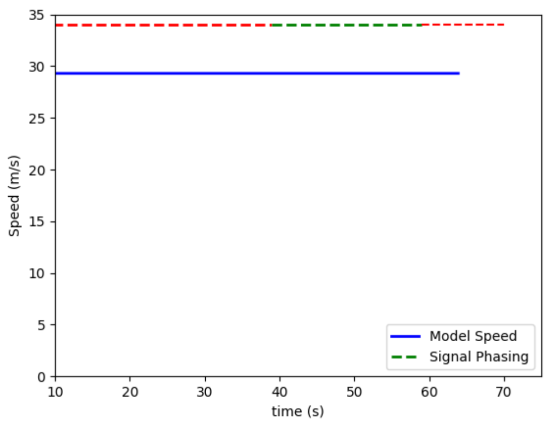
5.6.1 Green phase

When the target vehicle approaches an intersection, it is expected to proceed when the signal is in the green phase. This scenario assumes that the vehicle is traveling at its desired speed. The prediction indicates that the vehicle will reach the intersection while the signal is still in the green phase. Even when the vehicle is approaching at a low speed or accelerates while approaching the intersection, the algorithm is totally based on the speed at the current time step. First, it predicts the time when the vehicle will arrive at the intersection which is based on the current step. Second, if the signal status is green at the predicted time, it will not suggest any brake which means keeping the normal driver behavior. Therefore, the algorithm is adaptive based on the vehicle dynamics at the current time step.

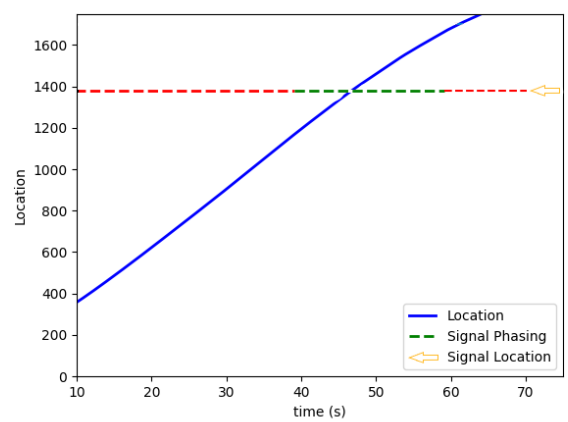
As a result, the warning algorithm will recommend no deceleration, signifying that the RLRWS is not taking any action and the color of the message is green. Figure 5.6 provides a detailed depiction of the vehicle's dynamics, including signal phasing, signal location, and warning messages. The vehicle will pass the intersection at a consistent speed, implying that the algorithm does not recommend braking, and the warning message remains green throughout. The current status of the traffic signal is depicted with the dashed line in all plots and the location of the traffic signal is marked with a yellow arrow in Figure 5.6b.

5.6.2 All red phase

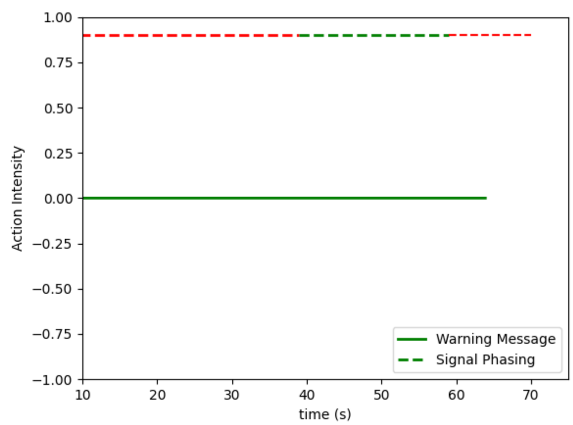
Approaching a red light necessitates coming to a complete stop before the stop bar. Consequently, a target vehicle, under the guidance of the warning algorithm, must halt in proximity to the stop bar for a single vehicle when the signal displays a red phase. The scenario of multiple vehicles in a platoon are not presented in this Task and



(a) Speed (m/s) vs. time(s)



(b) Trajectory



(c) Warning Message

Figure 5.6: Warning message, speed, and trajectory plot in respect to simulation time when approaching and passing a green light.

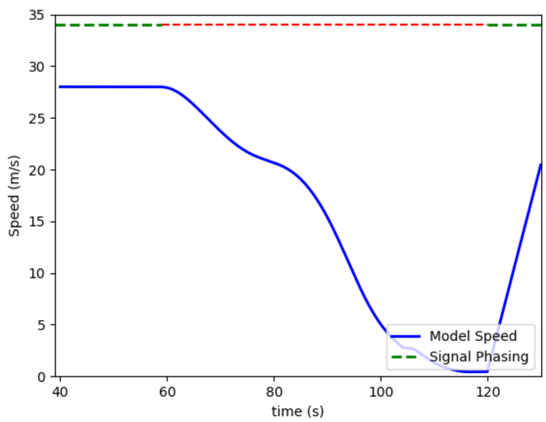
will be studied in the following tasks. A simpler scenario when approaching a signal involves driving toward an intersection that consistently remains in the red phase until the vehicle comes to a complete stop. In other words, both the current signal phase and the signal phase predicted for when the target vehicle reaches the intersection are red phases as long as the signal location is within the prediction horizon. This ensures that both the algorithm and the driver are aware that they need to decelerate.

Figure 5.7 illustrates the vehicle dynamics for this scenario. The target vehicle gradually decelerates and comes to a stop before the red line. When the signal changes to green, the driver resumes normal driving behavior and accelerates. As shown in Figure 5.7c, deceleration values start at a lower magnitude and increase as the vehicle approaches the signal. These values are determined through the optimization process described in Section 5.4 and suggested to the driver as the warning message. Concerning warning messages, as the deceleration value increases, the warning transitions from green to yellow, indicating a relatively higher deceleration rate. Even as the deceleration values decrease closer to the signal, the warning message still remains in the yellow color until the vehicle comes to a complete stop near the stop bar to prevent driver confusion.

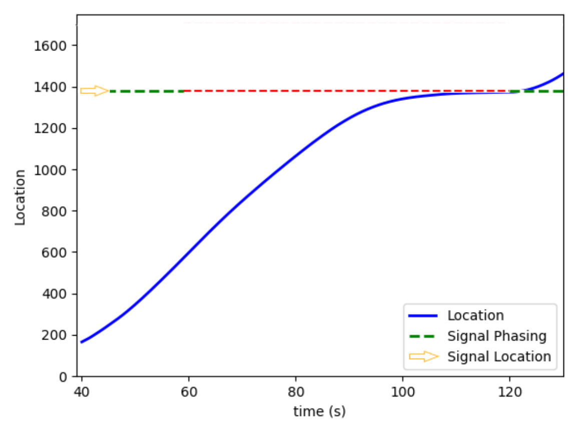
5.6.3 Switching from green to red phase

To encompass all potential scenarios, the project team has examined cases where the signal is already in a certain phase and changes to another phase as the vehicle approaches. When the current signal phase is red and will transition to green as the target vehicle approaches, there are no concerns related to red light running, as the vehicle will be able to proceed through the intersection during the green phase. However, when the current phase is green but is set to turn red as the vehicle approaches the stop bar, it represents a common scenario of red light running. The primary reason for this is that drivers only perceive the green signal at the intersection, unaware that it will soon turn red before they can safely cross. By the time it changes to red, it may be too late to brake and stop. Therefore, the warning algorithm, utilizing V2I connection, should offer an improvement when such signal transitions occur.

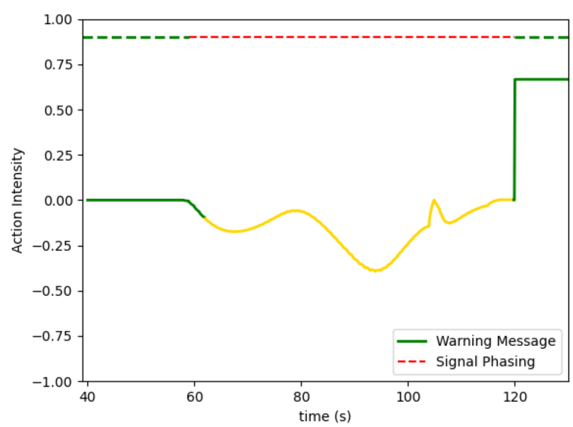
In this particular scenario, the signal changes from green to red when the signal location falls within the vehicle's prediction horizon. As depicted in Figure 5.8, at $t = 59$, the signal switches to red when the vehicle is approximately 250 meters away from the signal. Figures 5.8c and 5.8a illustrate that the target vehicle attempts to decelerate even when the signal is displaying a green phase. Even the driver was initially unaware of the impending red light, the SPaT messages reveal that the next signal status will be red in a certain time stamp and update the warning algorithm with this information. The target vehicle is modeled in SUMO by being aware of the current and upcoming signal phasing plans. During this time period, the warning message remains in the yellow phase, signaling the need to slow down. It is visible that several seconds before switching from green to red, the algorithm suggests slowing down because it has predicted that by the time the vehicle arrives at the signal location, the signal



(a) Speed (m/s) vs. time(s)



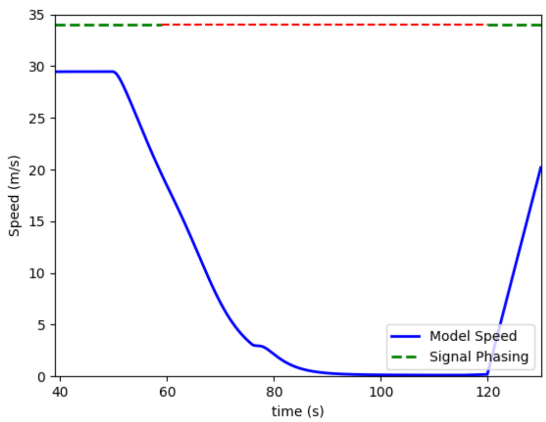
(b) Trajectory vs. time(s)



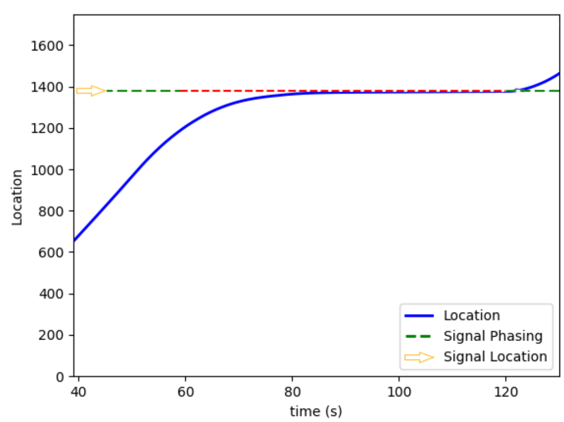
(c) Warning Message

Figure 5.7: Warning message, speed, and trajectory plot with respect to simulation time when approaching a red light.

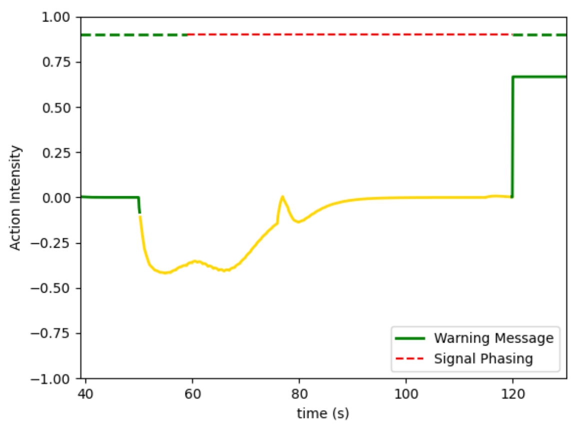
is red. Notably, the warning message is not red during this phase since the driver still has sufficient time to take corrective action even when the signal eventually turns red. Finally, the vehicle resumes acceleration as part of its normal behavior when the signal switches back to green.



(a) Speed (m/s) vs. time(s)



(b) Trajectory



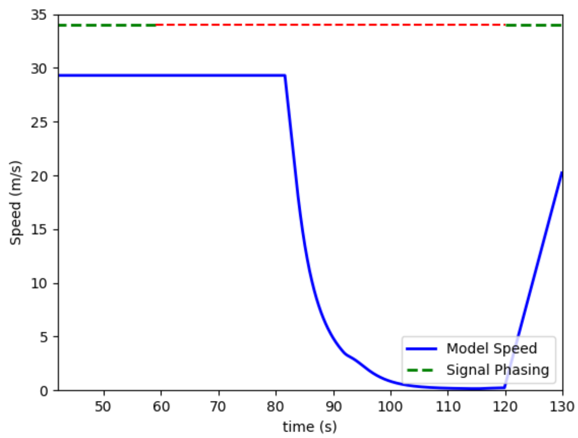
(c) Warning Message

Figure 5.8: Warning message, speed, and trajectory plot with respect to simulation time when approaching a green light but it turns to red.

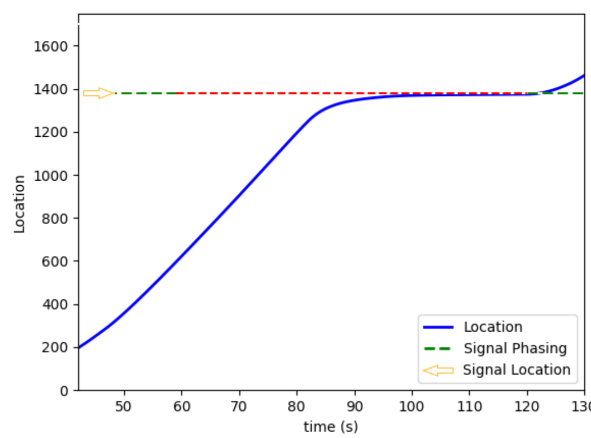
5.6.4 Warning message is followed tardily

As previously mentioned, there is a significant possibility that drivers may not adhere to the recommendations provided by the warning algorithm, and they might not even acknowledge the yellow warning message. Instead, they might continue driving until the last moments before making a sudden and more abrupt deceleration. The project team has modeled this scenario as an event in which the target vehicle maintains a constant speed up to a certain location. Consequently, the deceleration values prescribed by the warning algorithm are not followed until reaching this location. After passing this point, the vehicle attempts to adhere to the warning algorithm, which is expected to suggest higher deceleration values since it delays slowing down until the last moments. One potential scenario has been modeled in which the vehicle travels at its maximum speed until there are 130 meters remaining to the signal. Beyond that point, the vehicle is guided by the warning algorithm.

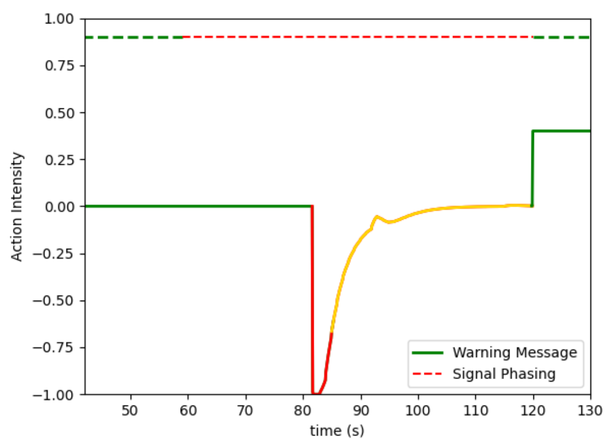
Figure 5.9 illustrates the described scenario. In this case, since the vehicle is in close proximity to the intersection, enabling the warning algorithm results in abrupt deceleration and a red light color. Generally, when braking occurs later, the deceleration rate tends to be higher compared to having more time to gradually slow down. In Figure 5.9c, a hard brake is suggested to reduce the speed with the red light, followed by a relatively lower brake rate with the yellow light as getting close to the intersection until stopping at the intersection. Finally, the vehicle returns to normal driving behavior as the signal turns green which means accelerates and passes the intersection.



(a) Speed (m/s) vs. time(s)



(b) Trajectory



(c) Warning Message

Figure 5.9: Warning message, speed, and trajectory plot with respect to simulation time when enabling the warning algorithm in the middle of the road (130 meters to signal).

5.7 Conclusion

This task is focused on modeling the red light running warning system for a single vehicle and integrating it into the microsimulation platform. The information received through V2I connectivity is used to generate essential data, such as the signal phasing plan, remaining distance to the signal, and vehicle dynamics. This prepared information is used to update a warning algorithm that aims to control the target vehicle, ensuring compliance with traffic constraints and enabling the vehicle to come to a complete stop before the stop bar when the signal is in the red phase. The algorithm predicts the vehicle's location, thereby determining the time step at which the vehicle will reach the intersection. The output of the algorithm is an acceleration-deceleration value used to control the target vehicle. The focus is on the deceleration process and guiding the driver to avoid harsh brakes by suggesting moderate braking.

To simplify and facilitate monitoring of the red light running process, the outputs are translated into warning messages for the deceleration process, including a bar for the action intensity with green, yellow, or red color for classifying. These messages signify the level of action, ranging from safe passage without any action to making a hard brake. Various scenarios are presented to cover all possible situations when a single vehicle approaches an empty intersection. Subsequent tasks will delve into traffic studies and the implementation of this method on actual vehicles, followed by road testing.

Chapter 6

RLRWS for vehicle in car-following

February, 2024

6.1 Introduction

The advent of connectivity has brought a new outlook to traffic engineering. Two crucial aspects of connectivity — vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication — have unlocked a world of opportunities. This innovation holds the promise of improving road safety, making energy consumption more efficient, and enhancing the overall flow of traffic [37] and [38].

In the initial tasks (Task 1 and Task 2), we have described the installation of essential equipment within both vehicles and at signalized intersections. The primary objective was to establish seamless communication between these critical components on the road, with a particular focus on receiving signal timing plans as vehicles approach intersections. Subsequently, the proposed red light running warning system (RLRWS) is presented and evaluated in the microsimulation testbed in Task 3 and Task 4. RLRWS is designed in order to guide drivers when a potential red light violation is noticed. The system utilizes connectivity both in vehicles and infrastructure. Task 4 presented RLRWS for an individual target vehicle and evaluated it in the microsimulation testbed. In the current task, RLRWS is implemented for a target vehicle when following other vehicles in the microsimulation testbed. Connected vehicles (CVs) in front of the target vehicle can transmit information to the target vehicle.

An individual vehicle case is defined as when there is no vehicle near the target vehicle. Thus, RLRWS demonstration for an individual vehicle does not need to anticipate the motion of the preceding vehicle. However, if the target vehicle is surrounded by other vehicles, particularly some CVs in front of the target vehicle, the warning algorithm considers the dynamics of those vehicles as well. Demonstration of the RLRWS in a platoon of vehicles is described in this task.

The proposed warning algorithm for red light running violations is converted into a real-time implementable system. RLRWS begins with receiving the necessary information through perception sensors, V2V and V2I connectivity. This information, which includes vehicle dynamics information and SPaT information, is continually updated as the target vehicle approaches the intersection. Vehicle dynamics generally encompass two aspects: First, the dynamics of the target vehicle, which includes its current speed and coordinates. Second, information about other vehicles, especially the preceding vehicle, such as speed and spacing to the target vehicle. This information is transmitted via V2V connections. Given more information from other CVs in the platoon, the target vehicle is able to predict the traffic states for the future which is discussed in this task. This opportunity to predict the traffic states considering other vehicles was not covered in Task 4. Based on the output of the prediction algorithm, the optimization-based warning algorithm will compute the optimal warning signal in real-time. Finally, a visualization function will be included to display the warning message on the screen in real-time.

Fig. 6.1 explains the aforementioned process. Measurement and connectivity obtain the required data and then update the algorithm which computes an adaptive warning message. The algorithm returns the results to the vehicle to be displayed and control the target CAV accordingly. The process is been repeated as a cycle for the next time step when updated information are available.

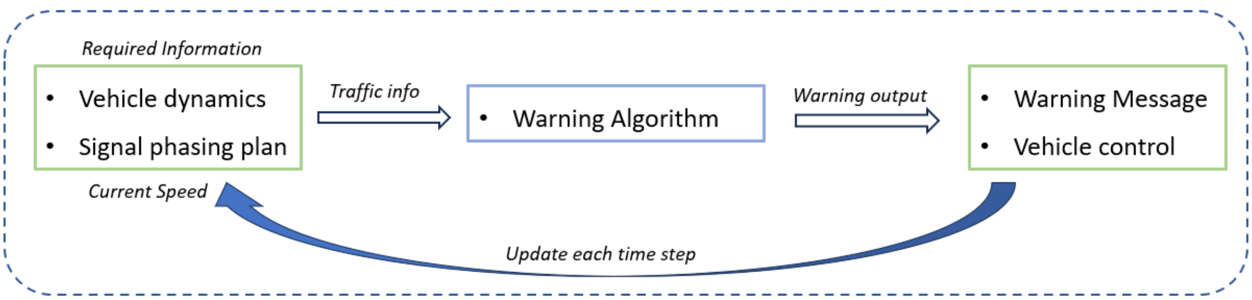


Figure 6.1: An overview of the RLRWS framework

It is crucial to test the proposed RLRWS before implementing it in vehicles and deploying it on the road. Consequently, the RLRWS is modeled and tested in microsimulation to ensure the correctness and reliability of the algorithm's performance. The Simulation of Urban MObility (SUMO) is utilized as the microsimulation tool to control a target vehicle by adhering to the warning messages generated by the algorithm. SUMO simulation is introduced in Task 3 in detail.

The warning message will be clear and user-friendly, consisting of a message that guides the driver on the appropriate action to enhance safety. Our ultimate goal is to make intersections safer through this innovative technology by guiding the driver when required. The structure of the current task begins with describing the warning message and control plan. Subsequently, the case of car-following for the target vehicle is illustrated, followed by an elaboration on the warning algorithm. Finally, multiple scenarios of a target vehicle following preceding vehicles when approaching the intersection are elaborated in order to ensure the performance of the warning system and represent the warning message accordingly.

6.2 Warning Message

The warning algorithm forms an integral part of this system, addressing a complex mathematical problem within the confines of real-world physical constraints. It seeks to derive optimal acceleration values that should guide the vehicle's behavior. However, it is essential to underscore that the target vehicles in this context are controlled by the drivers. The driver retains control of the vehicle at all times and the RLRWS displays messages to guide the driver to a safe stop. The format of the warning message is explained in Task 4 with full details. The current task just summarizes it as a reminder.

Within this framework, the warning messages are displayed in a colored circle where values represent the accelerate-decelerate intensity with respect to the maximum braking and maximum acceleration rate. Since the project is focused solely on studying the stopping process, the main focus is on deceleration and the acceleration process is studied as normal driving behavior (the driver has regular actions and no guidance is required). Therefore, the color mainly describes the suggested brake intensity. Colors are categorized into three distinct colors: Green, Yellow, and Red, as outlined in Table 6.1. Furthermore, Fig. 6.2 displays the full range of all possible warning messages with the corresponding color. For example, if the maximum deceleration is defined to be 5 meters per second squared, a warning message of 50% means to brake with half power or a rate of 2.5 meters per second squared. This example will be displayed as a yellow-colored message.

Table 6.1 and Fig. 6.2 explain the warning bar where the colors represent the intensity range. the green color represents normal driving behavior, and the RLRWS does not provide any additional guidance, encompassing all positive acceleration values as well as minor acceleration values. The yellow color indicates that the driver needs to make a moderate deceleration, including moderate deceleration rates (or moderate brake intensity rates), while the color switches to red when there is an imminent risk of a red-light violation with the necessity of an intense brake. Generally, the red color is displayed when the driver has not followed the yellow color bar displayed previously. As a result, yellow switches to red, and the brake intensity is increased in the displayed bar gradually as the driver did not follow the message.

In the case of acceleration, the system only displays "normal driving" with a green color without specifying the intensity amount. Fig. 6.3 illustrates three examples of what is displayed to the driver at each instance. The warning message is provided in a horizontal bar ranging from normal driving behavior to the maximum brake that

Table 6.1: Warning message color classification

Warning Message	Definition
Green	No action required, far from signal or signal is green when the vehicle arrives
Yellow	Driver needs to decelerate gradually, a full stop is required soon
Red	A hard brake is required, otherwise will run the red light

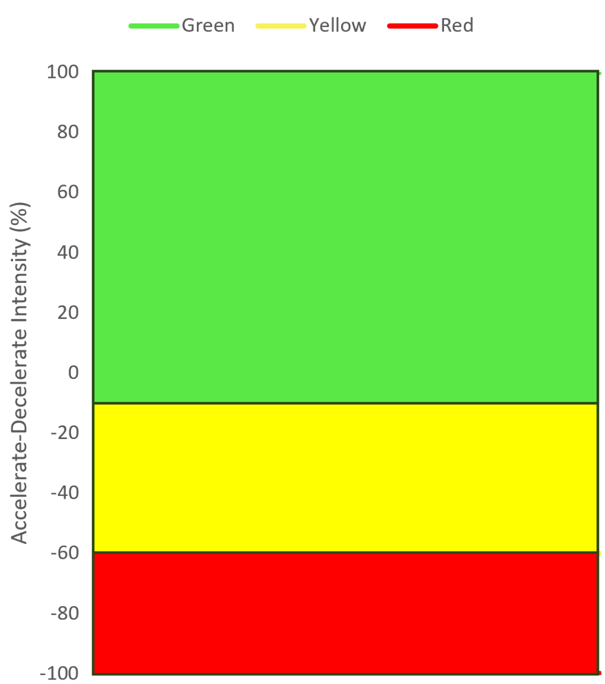
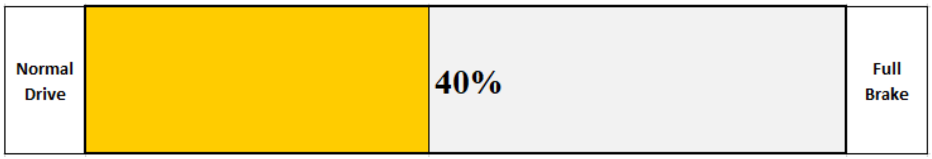


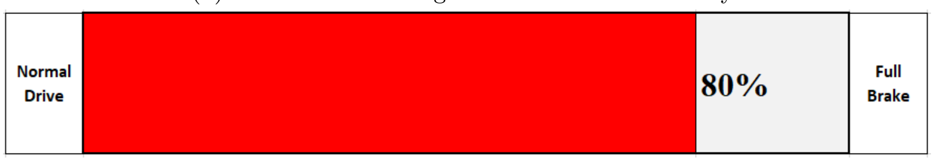
Figure 6.2: Warning message range, where the vertical axis represents the intensity of acceleration when above zero and deceleration when below zero

displays the suggested brake intensity at the current time. For example, Fig 6.3b and Fig. 6.3c also represent brake intensities with the corresponding value that comes from the algorithm. Since the warning message for these cases ranges between 0 and 100 percent, the bar spans between normal driving behavior and the maximum brake.

(a) Normal Driving, green color



(b) Yellow color message with 40% brake intensity



(c) Red color message with 80% brake intensity

Figure 6.3: Examples of warning message displayed to the driver while driving

Crucially, it should be emphasized that the warning message system operates on a time-dependent continuum. It begins with a Green color, signifying that all conditions are safe with regard to red-light violations. Subsequently, it transitions to Yellow, indicating the need for caution, and finally to Red if the driver fails to initiate sufficient deceleration for a safe stop. Thus, it never immediately starts by displaying a harsh brake intensity via red color. Only if the driver does not follow the message which means a red light violation is more possible, the bar intensity will increase to suggest a harsh brake. However, in typical driving scenarios where drivers maintain a deceleration intensity close to the suggested rate, the warning message will not escalate to the Red level.

6.3 Target vehicle in car-following scenario

Studying an individual target vehicle may not always be realistic. Vehicles are often traveling as a platoon with the surrounding traffic where they follow each other. Such behavior is modeled as car-following models in the studies that consider parameters such as spacing, and relative speed between the vehicle and its leading vehicle. In this task, the target vehicle is following other vehicles referred to as the preceding vehicles. The dynamics of the preceding vehicles are available in real-time as well as the dynamics of the target vehicle. Information from the preceding vehicles is obtained by V2V connectivity when the preceding vehicles are CVs. The behavior of the immediate preceding vehicle in front of the target vehicle can also be measured by the sensors installed on the target vehicle. For the experimental road test, the GPS antenna of the OBU device as described in Task 1 and

Task 2 will provide the target vehicle and other CVs' speed and location information in real time.

The received information from vehicle connectivity along with the SPaT information is employed to predict future traffic states. Given the predicted states, the target vehicle is aware of the road ahead while approaching the intersection. Utilizing this information in the warning algorithm is described in Sec. 6.4.

One of the major challenges for the RLRWS in car-following is when the platoon is approaching a green light and it turns red meanwhile. In this case, the preceding vehicle can cross through the intersection in green or yellow status but the target vehicle faces a red light. RLRWS should guide the driver in order to prevent red light running violation in this case.

6.4 Warning algorithm

In this section, we show the formulation of the proposed red light running warning system considering car-following scenario. In this task, the RLRWS system shown in Task 4 is modified to utilize information from vehicles in front of the target vehicle. Such information could become available through vehicle connectivity techniques or from sensors on the target vehicle. RLRWS in this task begins with an illustration for multi-vehicle scenarios followed by a model-predictive control (MPC) problem which computes the optimal warning message.

The proposed algorithm is formulated in MPC fashion, which calculates the optimal input of a control system under several constraints given the initial condition. The MPC algorithm can optimize the system's input based on the system's predicted future states. This is suitable for the development of the RLRWS, which requires the algorithm to consider the future behavior of a driver and future traffic conditions given the future signal status. Fig. 6.4 shows the general framework of the proposed system. The input of the system is the traffic condition, which includes the target vehicle's speed and location information, SPaT information from traffic signal that are in the V2I range, and other vehicles' speed and location information received through V2V communication or perception sensor. The output of the system is the warning signal, which suggests the target vehicle's driver to do reaction in advance to avoid running a red light. Two modules are included in the system: traffic prediction algorithm and MPC-based optimization problem. The proposed system will update the warning signal every 1 second and details about the system will be discussed in the following subsections.

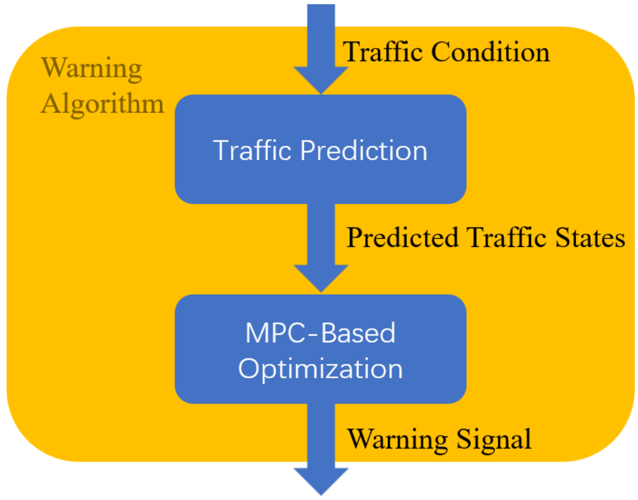


Figure 6.4: A framework of proposed red light running warning algorithm. The input of the system is the traffic condition and the output of the system is the warning signal to the vehicle's driver.

6.4.1 Traffic prediction algorithm

To provide the warning signal to the vehicle's driver in advance, the algorithm should have the capability to predict future traffic states. The traffic prediction framework developed in [39] is used in the proposed RLRWS system. Using the target vehicle's speed and location information, other CVs' speed and location information, along the SPaT information from nearby signalized intersections, the traffic prediction algorithm will predict the traffic states ahead of the target vehicle for the next 10 seconds. The road ahead of the target vehicle is divided into M cells of equal length and their traffic states are estimated by a state observer (Kalman filtering) using vehicle dynamics information [39]. Since the target vehicle and the target vehicle's preceding vehicle's speed profiles are relevant to the traffic states, their predicted longitudinal trajectory for the next 10 seconds can also be obtained. These predicted trajectories will be used by the MPC-based optimization problem.

6.4.2 MPC-based optimization problem

The following optimization problem is formulated for the RLRWS:

$$
\begin{aligned}
u_t^* = &\operatorname{argmin} \int_{t_0}^{t_f} q(x(t), v(t), a(t))dt && (6.1a)\\
\text{s.t.} \quad & \dot{x}(t) = v(t), && (6.1b)\\
& \dot{v}(t) = a(t), && (6.1c)\\
& a(t) = f(u(t), x(t), v(t)), && (6.1d)\\
& 0 \leq u(t) \leq 100 && (6.1e)\\
& v_{\min} \leq v(t) \leq v_{\max}, a_{\min} \leq a(t) \leq a_{\max}, && (6.1f)\\
& d(t) \geq d_{pre}(t) + \beta\sigma[d_{pre}(t)] - d_{max} && (6.1g)\\
& d(t) \leq d_{pre}(t) - \beta\sigma[d_{pre}(t)] - (d_{min} + h_{min}v(t)) && (6.1h)\\
& x(t) \leq x_{\text{tl}} - v(t)\tau_0, \text{if the traffic signal is red at } t, && (6.1i)\\
& v(t_f) = 0 \text{ and } x(t_f) \geq x_{\text{tl}} - d_0, \text{if the traffic signal} \\
& \qquad \text{is red at } t_f \text{ and } x_{\text{pred}}(t_f) \geq x_{\text{tl}} - d_0 && (6.1j)\\
& x(t_0) = x_{t_0}, v(t_0) = v_{t_0} && (6.1k)
\end{aligned}
$$

where $u(t)$ is the warning signal provided to the driver; $x(t)$, $v(t)$ and $a(t)$ are the vehicle's longitudinal position, longitudinal speed and longitudinal acceleration, respectively; t_0 and t_f represent the beginning and end time step of the optimization problem; x_{tl} is the longitudinal position of the traffic signal; τ_0 represents the desired time headway; d_0 is a small distance value; $x_{\text{pred}}(t_f)$ indicates the vehicle's predicted longitudinal location at the end time step of the optimization problem; $d_{pre}(t)$ is the predicted location of the preceding vehicle at time t from the traffic prediction algorithm described in Sec. 6.4.1; d_{max} is the upper bound for the distance between vehicles to ensure satisfactory and realistic traffic, d_{min} is the minimum following distance when the target vehicle is stationary; h_{min} is the time headway in seconds; $\sigma[d_{pre}(t)]$ represents the standard deviation of the estimated location of the immediate preceding vehicle to cover the uncertainties in the prediction and β determines the confidence level chosen to be 1 in numerical studies.

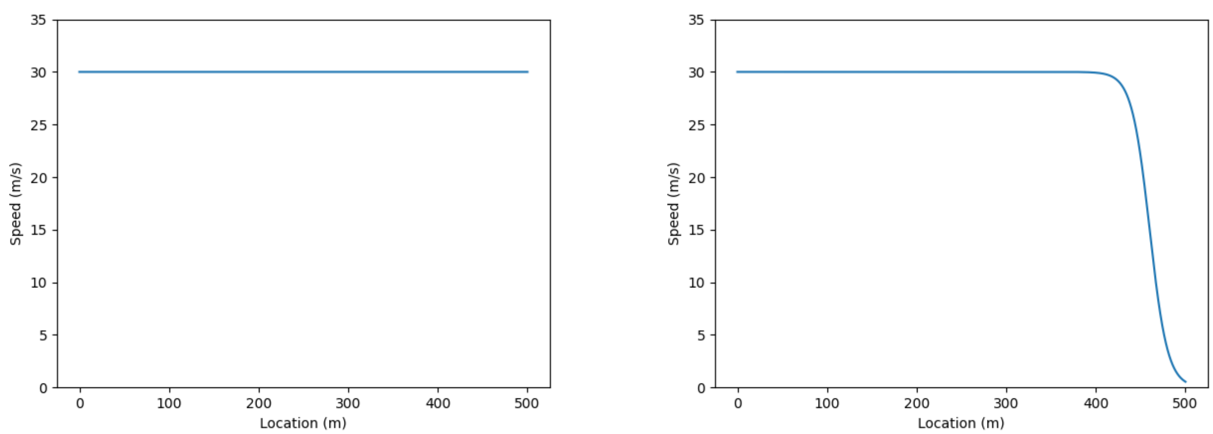
An explanation of equations is as follows:

- (6.1a) shows the optimization's objective function; function $f(\cdot)$ is the driver model, which shows the driver's reaction to the warning signal.
- (6.1e) represents the constraint on the warning signal's value.
- (6.1f) indicates the physical constraint on the vehicle's speed and acceleration.
- (6.1i) and (6.1j) are used to avoid running a red light and will be explained later.

- (6.1k) is the initial condition of the optimization problem.
- 6.1g and 6.1h represent the minimum and maximum allowed distance between the target vehicle and the preceding vehicle.

Equations 6.1g and 6.1h explain the car-following constraints in the presence of the preceding vehicle which is covered in this task. The target vehicle should follow the preceding vehicle according to these equations during the prediction horizon. Since the location of the preceding vehicle in the upcoming time steps is predicted in Section 6.4.1, the target vehicle is able to maintain a safe spacing to the preceding vehicle. The car-following constraints and employed parameters are thoroughly explained in [40]. This optimization problem could be solved using any type of nonlinear numerical solver.

The objective function is employed to describe the cost of a vehicle's movement along the prediction horizon. In this task, the objective function is set to $q(x(t), v(t), a(t)) = w_1 a^2(t) + w_2 \dot{a}^2(t) + w_3(v(t) - v_{\text{ref}})^2$, where w_1, w_2 and w_3 are weighting factors; v_{ref} is a reference speed as shown in Fig. 6.5. In the figure, we assume the traffic signal is at the 500 m. When the traffic signal is green, the reference speed is the free flow speed of the road. When the traffic signal is red, the reference speed drops before the stop bar. To simplify the computational process of the solver, this reference speed is modeled in the form of a sigmoid function. As a result, the proposed objective



(a) Reference speed when the traffic signal is green. (b) Reference speed when the traffic red is red

Figure 6.5: Reference speed used for MPC optimization.

function will minimize the vehicle's acceleration and jerk while tracking the reference speed along the prediction horizon, which provides a smoother longitudinal maneuver.

Given a warning signal, the driver will react to it by adjusting the vehicle's longitudinal speed $v(t)$. However, when the vehicle is driving at different speed or is locating at different longitudinal location, the driver's reaction to the same warning signal will be different. For example, when the vehicle is far away from a signalized intersection, the driver tends to apply a smaller force to the braking pedal compared to the scenario when the vehicle is close to the red light. Therefore, the real relationship between the warning signal and the vehicle's acceleration also depends on the vehicle's location and speed. In this work, the function $f(\cdot)$ is used to describe this. Similar to Task 4, this task uses a linear driver model $a(t) = -u(t)/20$ with $0 \leq u(t) \leq 100$ is used as this driver model. It is obvious that drivers may not follow this model perfectly and will have different behaviours. However, since the MPC problem will update the warning signal every 1 second, the algorithm will use the latest vehicle state at each update instance as the initial condition. For example, a larger warning signal will occur if the driver doesn't decelerate the vehicle at the beginning.

At the beginning of each optimization circle, the prediction algorithm will predict the vehicle's longitudinal trajectory along the prediction horizon. If the vehicle is predicted to reach the stop bar at the end of the prediction horizon t_f under red light (reach a certain range ahead of the stop bar, e.g. the vehicle is predicted to be within 20 m ahead of the stop bar at the terminal horizon), the constraint (6.1j) will be included in the optimization problem. This will force the vehicle to stop close to the stop bar at the time t_f. To avoid infeasibility of the problem, positive slack variables γ_v and γ_x are included in the optimization problem and constraint (6.1j) is re-written as: $v(t_f) == \gamma_v$ and $x(t_f) + \gamma_x \geq x_{\text{tl}} - d_0$. Term $w_v\gamma_v^2 + w_x\gamma_x^2$ with two large positive weighting factors w_v and w_x is added to the objective function (6.1a). In this way, the optimization problem allows the problem to slightly violate these constraints when they cannot be satisfied, which guarantees the feasibility of the algorithm.

When the ego vehicle is approaching the stop bar and the traffic signal is red, a shorter prediction horizon is required by the MPC optimization and the terminal constraint (6.1j) on distance will be re-written in a stricter form. This is because as the vehicle is approaching the intersection, it requires a shorter time to reach the stop bar and the vehicle is expected to stop closer to it. In this problem, the initial prediction horizon is 10 sec and d_0 is set to 20 m. When the ego vehicle is 60 m away from the stop bar, these two values will be updated to 10 sec and 15 m; When the ego vehicle is 40 m away from the stop bar, these two values will be updated to 8 sec and 15 m; When the ego vehicle is 20 m away from the stop bar, these two values will be updated to 6 sec and 5 m. This process will speed up the computational process and force the vehicle to stop before the stop bar completely.

6.5 Simulation results

This task aims to propose a warning algorithm that provides optimal deceleration rates and ensures a complete stop before the stop bar and for car-following scenarios. Therefore, this section primarily assesses the performance of the proposed algorithm for possible real-world scenarios. The signal phase when the vehicle platoon is approaching the intersection and when it arrives at the signal location is a key parameter. Therefore, the modeled scenarios include the green phase and red phase. The yellow light is considered an extension of the green light. The main reason for this convention is to set a time boundary that differentiates whether entering the intersection is permitted or not. That boundary is chosen as the light turning red because entering yellow is permitted if needed. It is feasible to adjust the boundary to be earlier (e.g. midway through the yellow light) but it runs the risk of being too conservative and thereby being ignored by drivers. In summary, this setup does not affect the stop-proceed decision and the vehicle will stop for the yellow light if it is possible. In terms of operation, When the vehicle is approaching the intersection while the signal is in the last few seconds of the green phase (which means it is in the yellow phase), RLRWS displays a warning message since the red phase is about to begin.

In the first scenario, the signal phase is consistently red when the target vehicle is approaching it. In this case, the warning system computes the optimal warning message to slow down the target vehicle smoothly before the stop bar. In the second scenario, the signal phase is consistently green when the target vehicle is approaching it. As a result, the warning system will not generate any warning messages. In the last scenario, the signal phase switches from green to red when the target vehicle is approaching the intersection. This is challenging for the warning system since the target vehicle's driver might run the red light during the switch process. As a summary, the traffic signal in respect to the signal will be as follows:

1. Consistent red light when the target vehicle is approaching in a vehicle platoon.
2. Consistent green light at first when the target vehicle is approaching.
3. Green light when the target vehicle is approaching that switches to a red light when the target vehicle arrives at the signal location. The target vehicle is following the preceding vehicle in this case where the preceding vehicle runs a yellow light but the target vehicle needs to stop at the red light.
4. Consistent red light when at first that switches to the green light but the target vehicle has to slow down due to congestion. This case represents when the target vehicle is predicted to travel at the free flow speed, but instead is forced to slow due to congestion in front.

These four cases can cover all potential cases in reality including traffic signal status and impacts of traffic states on the target vehicle. Every signal status is covered by the mentioned cases as well as changes in the speed of the vehicles ahead of the target vehicle which can be interpreted similarly to case 4.

In all the above scenarios, the target vehicle dynamics information along with its preceding vehicle dynamics information is available to the algorithm. This means the target vehicle's information, its preceding vehicle's information along with SPaT information will be used in the traffic prediction algorithm to predict the future traffic states. For each traffic case, three plots are presented including a space-time plot of the vehicle platoon, a speed plot, and an acceleration plot as the vehicle is approaching the intersection up to when the vehicle passes the intersection. The current status of the traffic signal is depicted with the dashed line in all plots and the location of the traffic signal is marked in all space-time plots. As described, the warning message is representative of the acceleration-deceleration values. Therefore, the raw warning message plot is not presented.

6.5.1 Case 1: all red phase

Approaching a red light necessitates coming to a complete stop before the stop bar. Consequently, the target vehicle, under the guidance of the warning algorithm, must halt in proximity to the stop bar for a single vehicle when the signal displays a red phase. Multiple vehicles in a platoon are presented in this task.

Figure 6.6 illustrates the vehicle dynamics for this scenario. The target vehicle (car_1 in the figure using black color) gradually decelerates and comes to a stop before the red line. When the signal changes to green, the driver resumes normal driving behavior and accelerates. As shown in Figure 6.6a, deceleration values start at a lower magnitude compared with its preceding vehicle (car_0 in the figure using blue color). These values are determined through the optimization process described in the previous section and suggested to the driver as the warning message. The vehicles platoon trajectory plotting shows that the warning algorithm allows the target vehicle to slow down more smoothly.

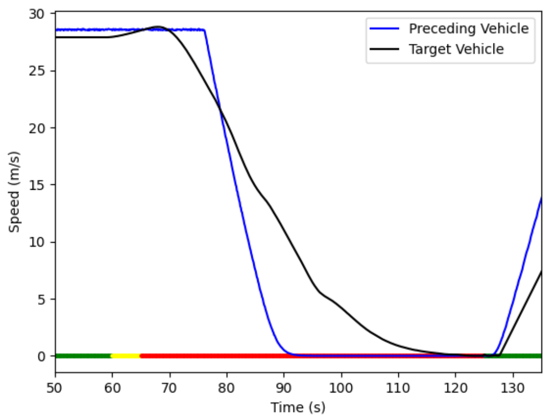
6.5.2 Case 2: all green phase

When the vehicle approaches an intersection, it is expected to proceed when the signal is in the green phase. This scenario assumes that all vehicles in the platoon are traveling at its desired speed. The prediction algorithm indicates that the target vehicle will reach the intersection while the signal is still in the green phase. First, it predicts the time when the target vehicle will arrive at the intersection which is based on the current step. Second, if the signal status is green at the predicted time, it will not suggest any braking which means keeping the normal driver behavior. Therefore, the algorithm is adaptive based on the vehicle dynamics at the current time step.

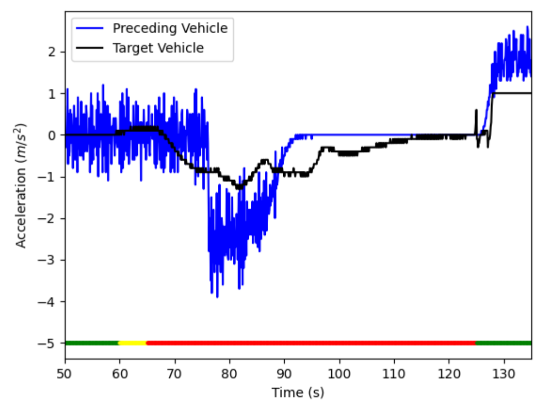
As a result, the warning algorithm will recommend no deceleration, signifying that the RLRWS is not taking any action and the color of the message is green. Figure 6.7 provides a detailed depiction of the vehicle's dynamics, including signal phasing, signal location, and warning messages. The target vehicle will pass the intersection at a consistent speed, implying that the algorithm does not recommend braking, and the warning message remains green throughout.

6.5.3 Case 3: switch from green phase to red phase

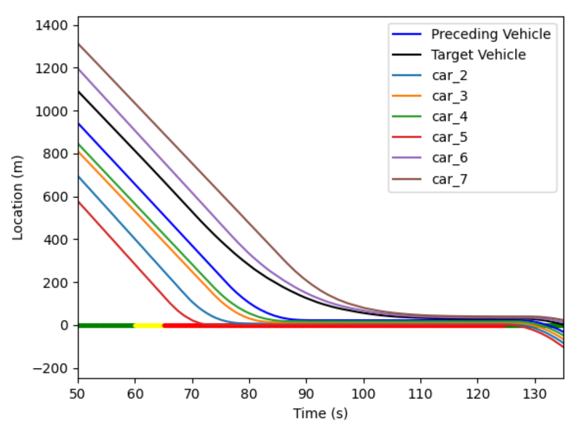
To encompass all potential scenarios, the project team has examined cases where the signal changes from the green phase to the red phase as the vehicle platoon approaches. When the current phase is green but is set to turn red as the platoon approaches the stop bar, it represents a common scenario of red light running. Especially, when the target vehicle's preceding vehicle passes the stop bar during the yellow phase, there might not be enough remaining duration for the target vehicle to pass the stop bar. A space-time plot of the vehicle platoon, a speed plot, and



(a) Target vehicle and its preceding vehicle speed profiles.

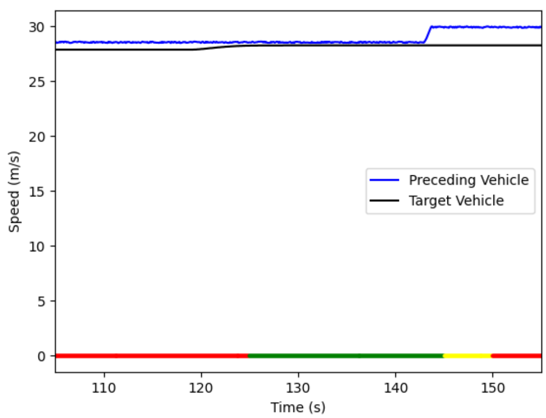


(b) Target vehicle and its preceding vehicle acceleration profiles.

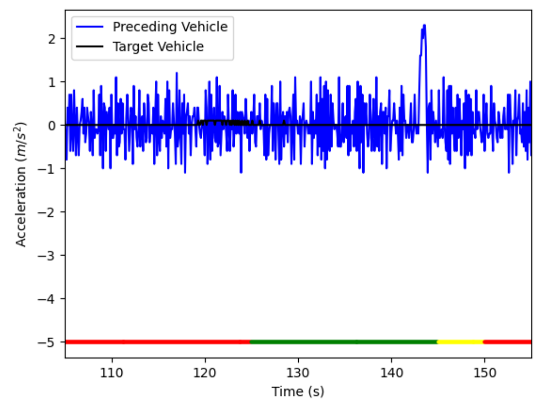


(c) Vehicle platoon trajectory

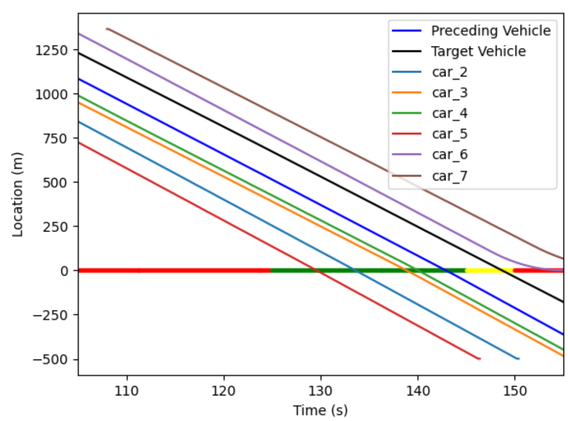
Figure 6.6: Speed, acceleration, and trajectory plot with respect to simulation time when approaching a red light.



(a) Target vehicle and its preceding vehicle speed profiles. The preceding vehicle (car_0) accelerates to a higher speed after passing the intersection due to an increase of speed limit.



(b) Target vehicle and its preceding vehicle acceleration profiles.

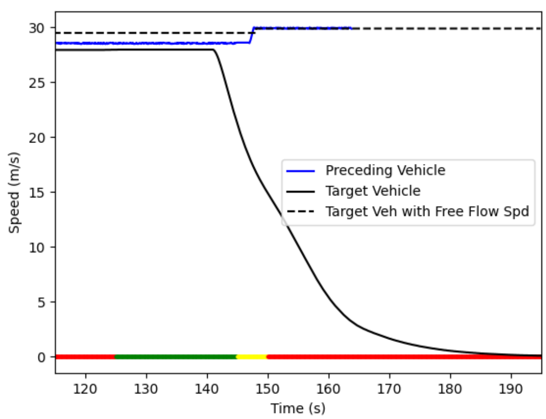


(c) Vehicle platoon trajectory

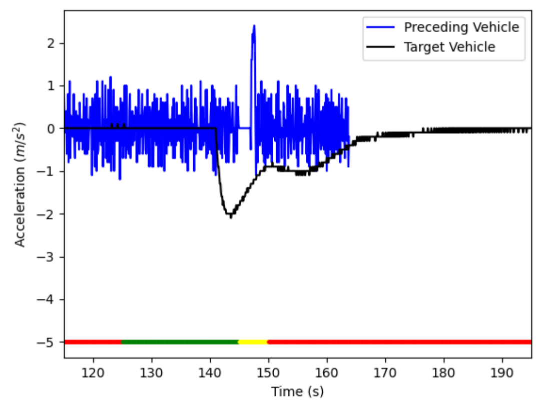
Figure 6.7: Speed, acceleration, and trajectory plot with respect to simulation time when approaching a green light.

an acceleration plot are shown in Fig. 6.8. To show the performance of the proposed RLRWS, the dashed black line in Fig. 6.8c indicates the target vehicle's trajectory when it drives at the road's free flow speed without other preceding vehicles.

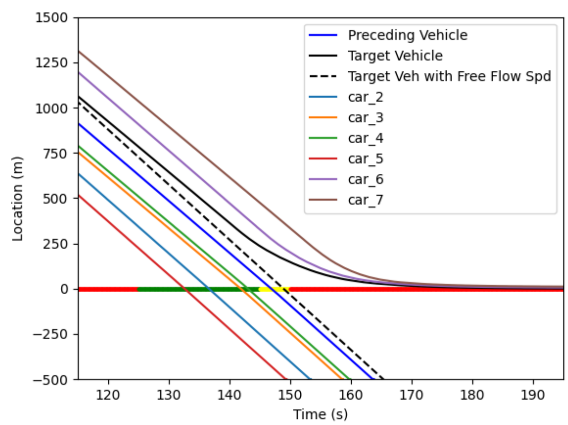
In this particular scenario, the signal changes from green to red when the signal location falls within the vehicle's prediction horizon. As shown in Fig. 6.8c, the target vehicle can pass the intersection before the red phase if there are no other preceding vehicles. In reality, however, it needs to stop before the stop bar due to the impact of the vehicle platoon. The warning algorithm recommends a smaller deceleration when the target vehicle is far away from the intersection. Therefore, the target vehicle slows down more smoothly and does not run the red light.



(a) Target vehicle, its preceding vehicle speed profiles along with target vehicle's speed profile when it drives at the free flow speed. The preceding vehicle (car_0) drives out of the modelled network after 170 s.



(b) Target vehicle and its preceding vehicle acceleration profiles. The preceding vehicle (car_0) drives out of the modelled network after 170 s.



(c) Vehicle platoon trajectory. The dashed black line indicates the target vehicle's trajectory when it drives at the road's free flow speed without other preceding vehicles.

Figure 6.8: Speed, acceleration, and trajectory plot with respect to simulation time when approaching a traffic signal changing from green to red.

6.5.4 Case 4: Slower flow due to congestion at green phase

In the previously mentioned cases, there was no vehicle queue at the signal location. The target vehicle can travel at the free flow speed if it arrives at a green light without congestion. In real cases, vehicles that were stopped in a queue during the red light begin to pass the signal location in order. Depending on the number of vehicles in the queue, it takes some time for the intersection to clear, and before that, any new arrival vehicle has to join the slower-moving queue.

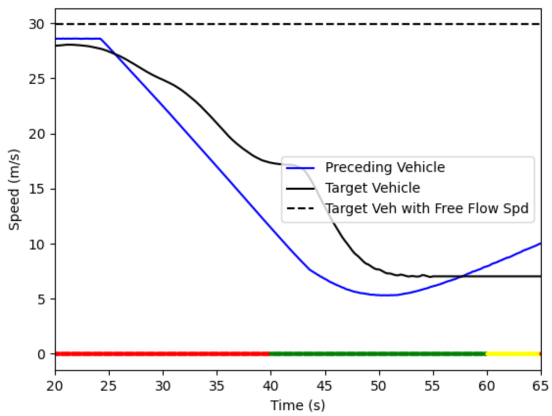
Case 4 represents when the target vehicle approaches a signal that has not yet dissipated. Although the target vehicle is predicted to travel at the free flow speed when the light is green, it should pass the intersection with lower speed values due to the congestion at the signal location. A space-time plot of the vehicle platoon, a speed plot, and an acceleration plot for this case are shown in Fig. 6.9. In this scenario, only the dynamics of the target vehicle and the preceding vehicle are assumed to be available. Therefore, RLRWS is working based on limited information about the platoon. To show the performance of the proposed RLRWS, the dashed black line in Fig. 6.9c indicates the target vehicle's trajectory when it drives at the road's free flow speed without other preceding vehicles. In this case, five vehicles have arrived at the red light and are leaving the signal location with lower speed values when the target vehicle arrives. Monitoring the trajectory of the preceding vehicle leads the target vehicle to slow down even when a green light is displayed and a free flow is expected. As shown in Fig. 6.9a, the target vehicle also slows down and passes the green light at a lower speed. This case can be compared to Case 2, particularly Fig. 6.7a where the target vehicle passes the signal location with the free flow speed.

Other situations where traffic in front of the target vehicle is not traveling with the free flow speed such as a slower truck passing the intersection or when a vehicle has stopped at the green light waiting for a left-turn yield can also be represented and modeled similarly to this case. In all of these cases, the target vehicle begins with a speed prediction equal to free flow speed but information from the preceding vehicles and the MPC-based warning algorithm results in a safe deceleration and passing the intersection at a slower speed.

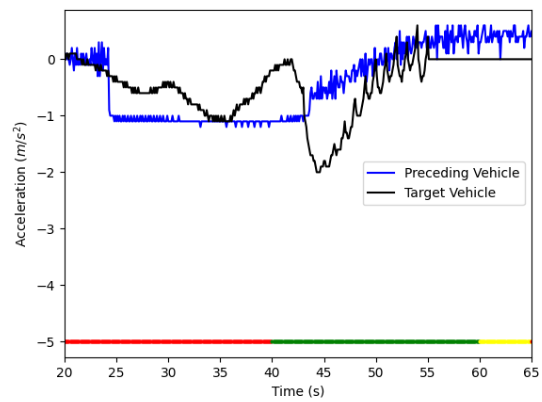
6.5.4.1 Case 4 with various connectivity rates in the platoon

As discussed, Fig. 6.9 represents a case where only the dynamics of the target vehicle and the preceding vehicle are available. In other words, all other vehicles in front of the preceding vehicle (vehicles c_0 to c_3) are not connected. This case illustrates the performance of RLRWS with limited connectivity data. However, a portion of the platoon might be CV that can transmit information to the target vehicle. This can improve traffic prediction accuracy described in Section 6.4.1.

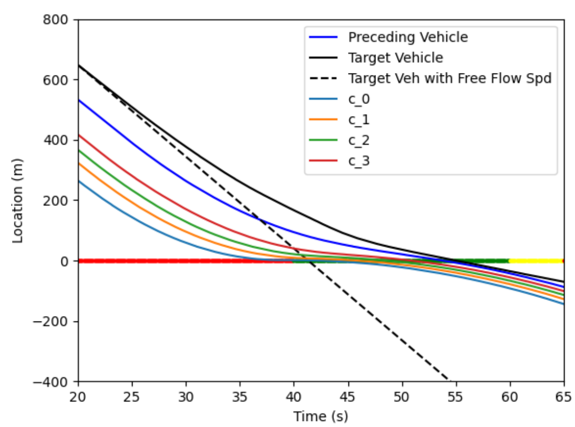
Fig. 6.10 represents a scenario from case 4 where two vehicles in front of the target vehicle are CVs in addition to the target vehicle. Similarly, Fig. 6.11 represents a fully connected platoon where the dynamics of



(a) Target vehicle, its preceding vehicle speed profiles along with target vehicle's speed profile when they approach a congested intersection.



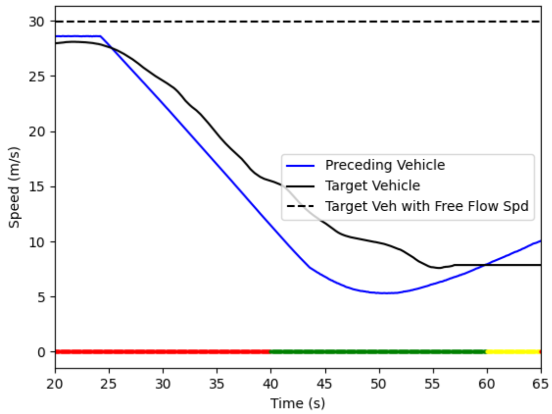
(b) Target vehicle and its preceding vehicle acceleration profiles.



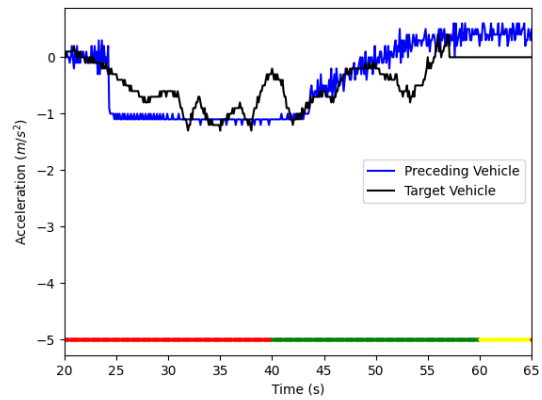
(c) Vehicle platoon trajectory. The dashed black line indicates the target vehicle's trajectory when it drives at the road's free flow speed without other preceding vehicles.

Figure 6.9: Speed, acceleration, and trajectory plot with respect to simulation time when approaching a green light but the target vehicle is forced to slow down.

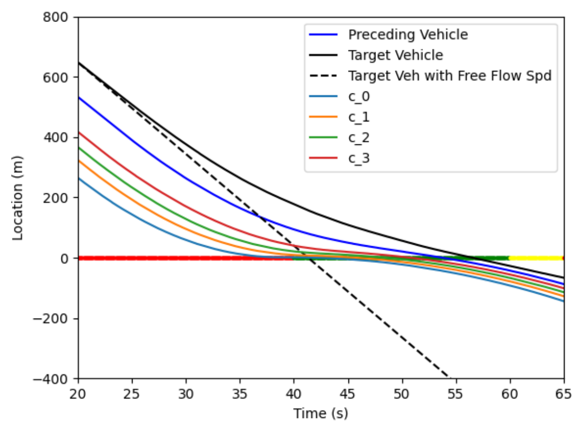
all vehicles are available for the target vehicle. Although the target vehicle heavily relies on the trajectory of the preceding vehicle, adding more CVs increases the spacing and improves speed smoothness resulting in safety enhancement.



(a) Target vehicle, its preceding vehicle speed profiles along with target vehicle's speed profile when they approach a congested intersection.



(b) Target vehicle and its preceding vehicle acceleration profiles.

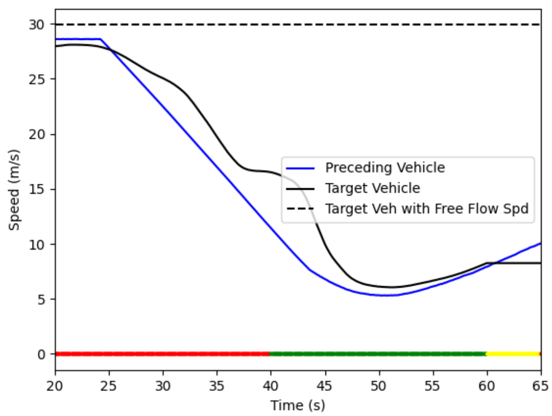


(c) Vehicle platoon trajectory. The dashed black line indicates the target vehicle's trajectory when it drives at the road's free flow speed without other preceding vehicles.

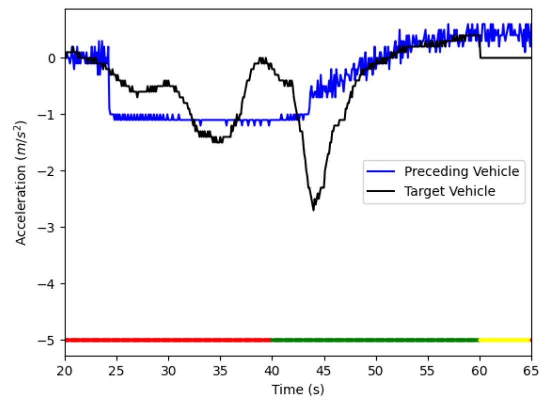
Figure 6.10: Speed, acceleration, and trajectory plot with respect to simulation time when approaching a green light but the target vehicle is forced to slow down where c_0 and c_2 are also connected vehicles.

6.6 Summary

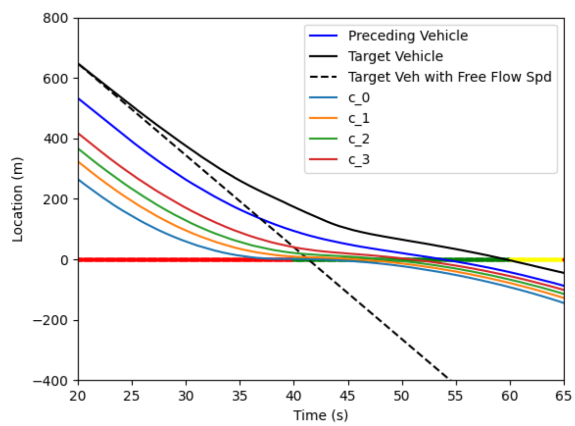
This task is focused on modeling the red light running warning system when the target vehicle is following other vehicles and integrating it into the microsimulation platform. The information received through V2I connectivity is used to generate essential data, such as the signal phasing plan, remaining distance to the signal, and vehicle dynamics. This prepared information is used to update a warning algorithm that aims to control the target vehicle,



(a) Target vehicle, its preceding vehicle speed profiles along with target vehicle's speed profile when they approach a congested intersection.



(b) Target vehicle and its preceding vehicle acceleration profiles.



(c) Vehicle platoon trajectory. The dashed black line indicates the target vehicle's trajectory when it drives at the road's free flow speed without other preceding vehicles.

Figure 6.11: Speed, acceleration, and trajectory plot with respect to simulation time when approaching a green light but the target vehicle is forced to slow down where all vehicles are connected in the platoon (100% penetration rate).

ensuring compliance with traffic constraints and enabling the vehicle to come to a complete stop before the stop bar when the signal is in the red phase. The algorithm predicts the vehicle's location, thereby determining the time step at which the vehicle will reach the intersection. The output of the algorithm is an acceleration-deceleration value used to control the target vehicle. The focus is on the deceleration process and guiding the driver to avoid harsh brakes by suggesting moderate braking.

To simplify and facilitate monitoring of the red light running process, the outputs are translated into warning messages for the deceleration process, including a bar for the action intensity with green, yellow, or red color for classifying. These messages signify the level of action, ranging from safe passage without any action to making a hard brake. Various scenarios are presented to cover all possible situations when a target vehicle approaches an intersection following other vehicles. Particularly, in the critical case when the preceding vehicle can cross the intersection but the signal turns green when the target vehicle arrives at the signal location is elaborated to ensure a safe stop guidance. Subsequent tasks will delve into traffic studies and the implementation of this method on actual vehicles, followed by road testing.

Chapter 7

Implement RLRWS with audio/visual warning for the on-road testbed

December, 2023

7.1 Introduction

In previous tasks, a microscopic traffic simulator was used to validate the proposed red light running warning system. In this specific task, the proposed warning algorithm is converted into a real-time implementable system. For real-time computation of the warning signal, the proposed system must have the capability to receive and interpret the connected vehicles messages e.g. signal phase and timing (SPaT) from the roadside unit and basic safety messages (BSM) in real-time. Subsequently, a traffic prediction algorithm is employed to predict traffic conditions at each time step. Based on the output of the prediction algorithm, the optimization-based warning algorithm will compute the optimal warning signal in real-time. Finally, a visualization function will be included to display the warning message on the screen in real-time.

The research team installed the roadside unit (RSU) at the intersection of Scott County's CSAH 18/CSAH 21/Southbridge Parkway with the help of MnDOT in Task 1. We validated the connectivity and reliability of the purchased RSU and OBU in Task 2. Microscopic traffic simulator and structure of the warning system are shown in Task 3 and Task 4, respectively. In this task, we will implement the real-time red light running warning system on a test vehicle. Further details about Task 6 will be discussed in the following sections.

7.2 Real-time warning algorithm framework

7.2.1 Hardware equipment

In this subsection, the hardware used for this task along with their connection is introduced. To broadcast the SPaT information of the traffic signal, the RSU is connected to the traffic signal controller as shown in Fig. 7.1. The RSU broadcasts the SPaT message via C-V2X communication. The socket in the traffic signal cabinet is used to supply power to the RSU.

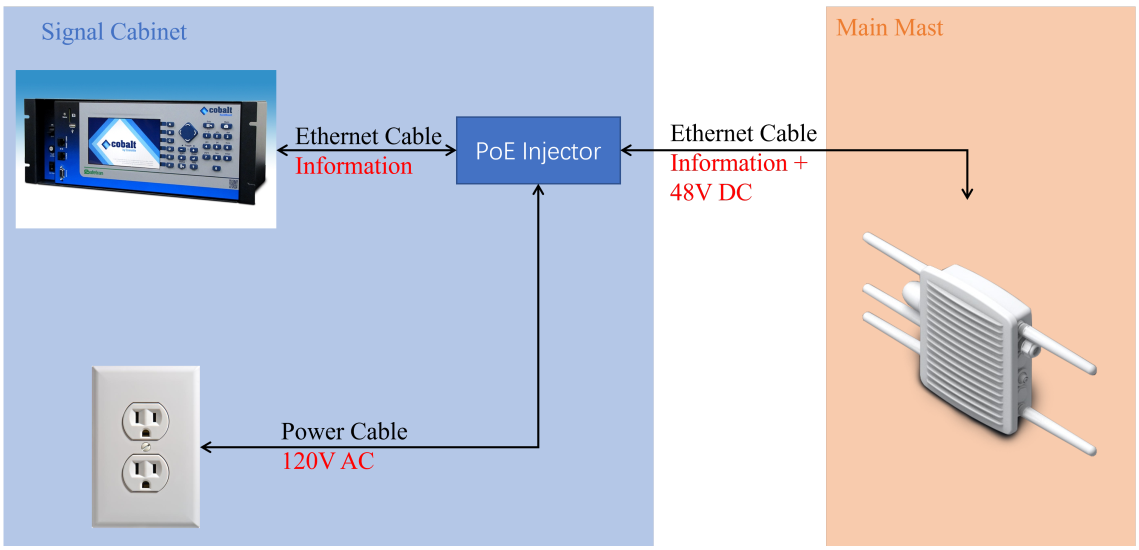


Figure 7.1: RSU configuration at the intersection.

The laptop and OBU are connected via an Ethernet cable in the test vehicles, where the OBU is powered by the vehicle's 12V DC power (see Fig. 7.2 and Fig. 7.3). In the figure, the green circle indicates the vehicle 12V DC power supply. The red circle shows the antenna of the OBU. The blue circle shows the OBU processor. The purple circle represents the laptop used in this project. The OBU receives the SPaT and BSM messages and sends them to the laptop in real-time via user datagram protocol (UDP). SPaT messages introduce the signalized intersection's traffic signal status. BSM messages contain the target vehicle's speed and location information. Utilizing the received data, scripts running on the laptop compute the optimal warning signal. Moreover, the laptop functions as the human-machine interface (HMI), displaying the warning signal on its screen. Ultimately, the driver can react to the warning message.

Figure 7.2: OBU and HMI configuration.

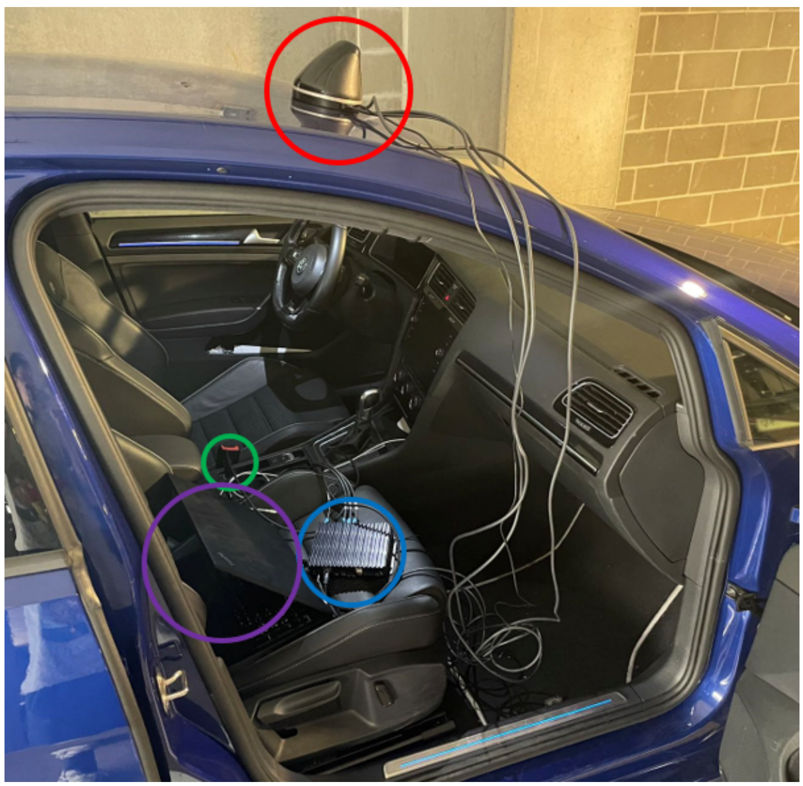


Figure 7.3: OBU installation for the test vehicle.

7.2.2 Software framework

In this subsection, the software framework (Fig. 7.4) of the proposed real-time warning algorithm is presented. The developed framework employs Python as the programming language. To enhance computational speed, the entire warning algorithm framework is divided into four individual blocks, which will run in parallel:

- OBU information process
- Traffic prediction

- Warning algorithm
- Visualization

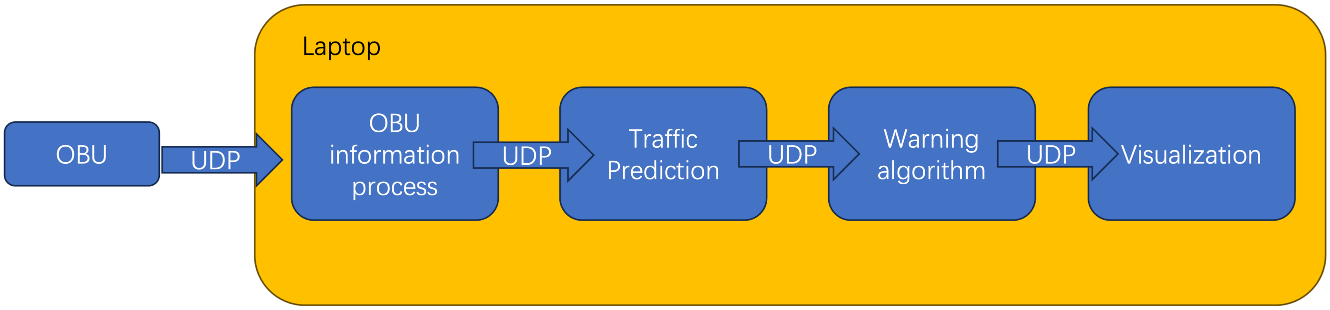


Figure 7.4: Software component of the real-time implementation of the proposed warning algorithm.

As demonstrated in the previous subsection, the UDP communication protocol facilitates real-time communication between the laptop and OBU. The UDP also serves as the communication standard among different blocks. UDP, known for its suitability in time-sensitive applications, enables faster communication since it does not spend time forming a firm connection with the destination before transferring the data. This attribute aligns well with the demands of developing this real-time warning algorithm.

For each programming block, a virtual buffer will be created to store received data from other programming blocks. These scripts will then read the most recent data from the buffer, ensuring access to the latest real-time system information. Further details about each programming block will be illustrated in the following subsections.

7.2.2.1 OBU information process block

This block receives and interprets the information transmitted from the OBU, and broadcasts the interpreted data to other programming blocks. To facilitate real-time broadcasting of the received message from the OBU to the laptop, the configuration file shown in Fig. 7.5 is stored in the device. The destination IP address and destination port number are specified in the configuration file. To receive the transmitted raw data stream in real-time, the "socket" package of Python [41] is used. The python package "pycmssdk" developed by Commsignia [30] is used to interpret the messages in real-time. Subsequently, these interpreted messages are broadcast to other programming blocks via UDP communication.

7.2.2.2 Traffic prediction block

This block utilizes real-time traffic information to predict traffic states along the prediction horizon. The traffic prediction algorithm developed in [39] without considering lane changing maneuvers is used in this project and its framework is shown in Fig. 7.6. In the figure, the red vehicle is the ego vehicle equipped with the proposed warning algorithm. The road ahead of the ego vehicle is divided into M cells of equal length. V2V, V2I communication

```json
{
    "enabled": true,
    "filters": [
      {
        "direction": "In"
      },
      {
        "radioInterface": 1
      }
    ],
    "out": {
        "format": "Raw",
        "udp":{
          "destHost": "192.168.0.67",
          "destPort": 8000
        }
    },
    "source": [
      "Wsmp",
      "Gnp"
    ],
    "version": 6
}
```


Figure 7.5: Configuration file in JSON format for real-time data transmission from Commsignia OBU.

and perception technologies are used to obtain real-time traffic information, which is used by a state estimator to estimate the traffic conditions of each cell.

The project intersection utilizes an actuated signal timing plan that does not follow a fixed phase duration. It extends the phase duration according to vehicle actuation. It begins with a minimum value for the current phase; when a vehicle is actuated near the ending point of the phase, the phase receives an extension. If vehicles constantly approach the signal during this time, this process continues until it reaches a predefined maximum time. As a result, the ”min time to change” used in this task is subject to change.

To deal with the actuated signal used at the intersection, the variable “min time to change” is used in this task as it’s close to the actual signal timing during the last few seconds of each signal phase. The variable completely accounts for the actuated signal and will be increased in the case of an extension. Furthermore, even though the variable ”min time to change” is subject to change, the remaining time interval is sufficient for the RLRWS algorithm. The reason is that the only important phase in terms of red light running is when the green light turns yellow and then red. In this case, only the green phase is subject to extension and the yellow phase has a fixed time without extension. Therefore, the remaining time to provide a safe stop considering the yellow phase is sufficient for a target vehicle.

Moreover, in the case of approaching a red light, the extension is not against the RLRWS algorithm since the algorithm correctly suggests a safe stop. Finally, the future traffic states can be obtained by propagating the traffic flow model (7.1).

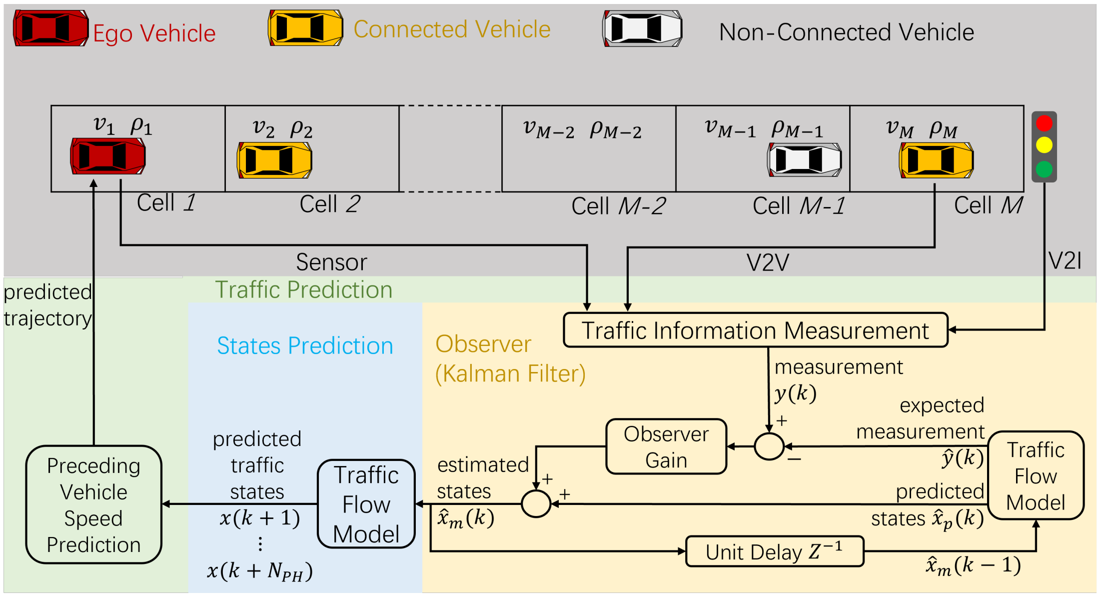


Figure 7.6: Traffic prediction framework used in the warning algorithm.

$$\rho_j(k+1) = \rho_j(k) - \frac{dt}{dx}\left[\rho_j(k)v_j(k) - \rho_{j-1}(k)v_{j-1}(k)\right] + \omega_j(k), \tag{7.1a}$$

$$v_j(k+1) = v_j(k) - \frac{dt}{dx}v_j(k)[v_j(k) - v_{j-1}(k)] + \underbrace{dt \cdot \frac{[\hat{V}_e(\rho_j(k)) - v_j(k)]}{\tau}}_{\text{Speed adaptation}}$$
$$\underbrace{- \frac{dt}{dx} \cdot \frac{c_0^2 \cdot [\rho_{j+1}(k) - \rho_j(k)]}{\rho_j(k) + \epsilon}}_{\text{Traffic pressure}} + \xi_j(k), \tag{7.1b}$$

where (7.1a) and (7.1b) describe the evolution of traffic density and traffic speed, respectively. k is the discretized time instance; dt denotes the time step size; dx represents the length of each cell; j is the cell index; ϵ is a small positive number to prevent zero denominator; $\omega_j(k)$ and $\xi_j(k)$ describe model uncertainties assumed to follow a Gaussian distribution; $\hat{V}_e(\rho_j(k))$ signifies the equilibrium speed of cell j; c_0 characterizes traffic pressure and τ describes the adaptation rate to reach the equilibrium speed.

Supposing an intersection is located at cell sc, then speed of this cell is set to zero when the traffic signal is red:

$$v_{sc}(k+1) = \begin{cases} 0, & \text{signal is red} \\ \text{right side of (7.1b)}, & \text{signal is green} \end{cases} \tag{7.2}$$

To facilitate real-time implementation, different from the work in [39], the discretization time dt has been adjusted from 0.1s to 0.2s. The prediction horizon is selected as 10s. The algorithm will update the traffic prediction output every 0.2s.

7.2.2.3 Warning algorithm block

This block uses real-time traffic information and the latest traffic prediction results to compute the optimal warning signal. As shown in the previous task, the proposed algorithm is formulated in the model predictive (MPC) fashion. To obtain the numerical solution for the formulated optimization problem, a time step with 0.2s is employed to discretize the problem. The predictive horizon is set to 10s. It is solved using IPOPT [42] with CasADi [43] used as modeling language. Considering that the computational time for solving the optimization problem is about 0.45s for each time step and humans cannot change behavior very often, the algorithm will update the warning message every 1s.

7.2.2.4 Visualization block

To visualize the warning message, the "matplotlib" package is employed to display the warning message through a figure on the laptop screen. As demonstrated in the previous task, three different colors-green, yellow and red-along with varying sizes are used to display the warning signal. In this task, the warning message is displayed with colored circles (see Fig. 7.7 and Fig. 7.8), where the diameter change represents the required intensity, such as a large red circle meaning a harsh brake is suggested. A colored circle performs faster visualization process compared to a more detailed horizontal percentage bar.

The visualization block updates the output upon receiving the newest warning message.

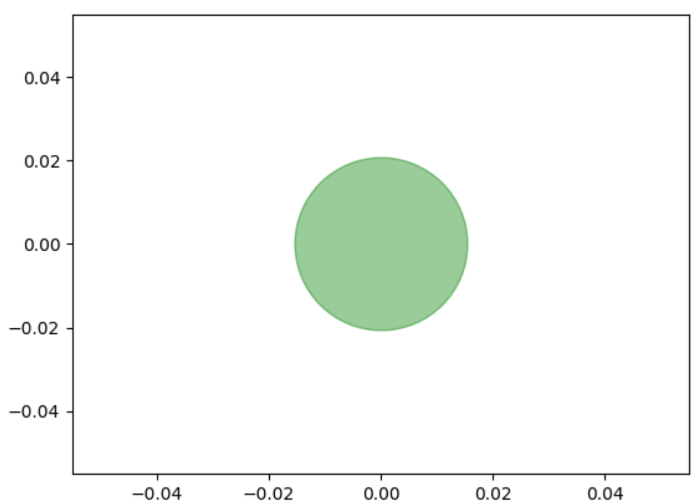


Figure 7.7: Warning output example 1.

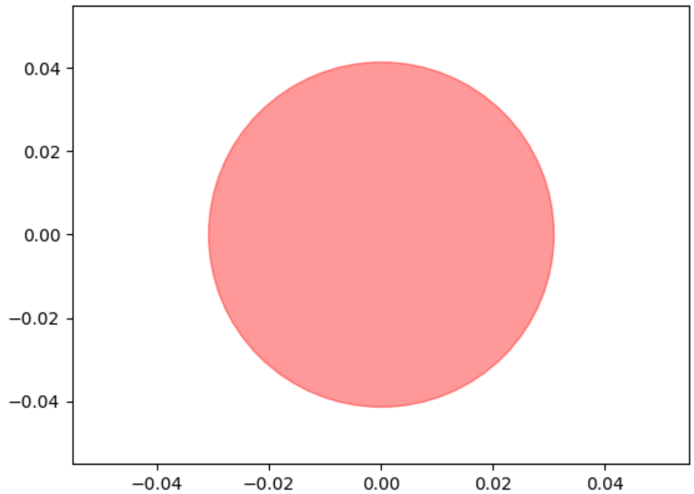


Figure 7.8: Warning output example 2.

7.3 Result

In December 2023, the research team went to the Scott County's CSAH 18/CSAH 21/Southbridge Parkway intersection to conduct testing on the proposed real-time warning framework. The team tested the algorithm in real-time and recorded relevant data on the laptop for subsequent analysis.

During the test drive, the test vehicle traveled northbound on CSAH 21 toward the signalized intersection. A research team member in the passenger seat observed the screen and communicated the warning message to the driver. The Warning message along with vehicle's speed, trajectory and acceleration profiles (Fig. 7.9–Fig. 7.12) from a test drive where the test vehicle approached the intersection during a red light phase is used for analysis in this section.

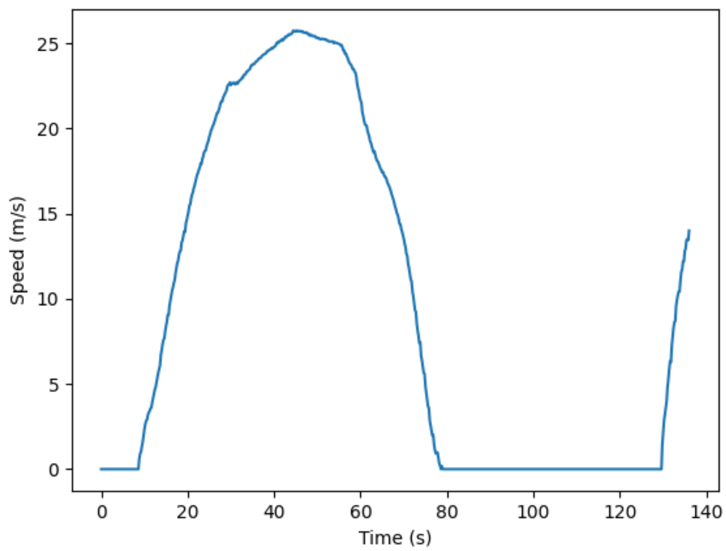


Figure 7.9: Target vehicle's speed when it drove towards the intersection during red light.

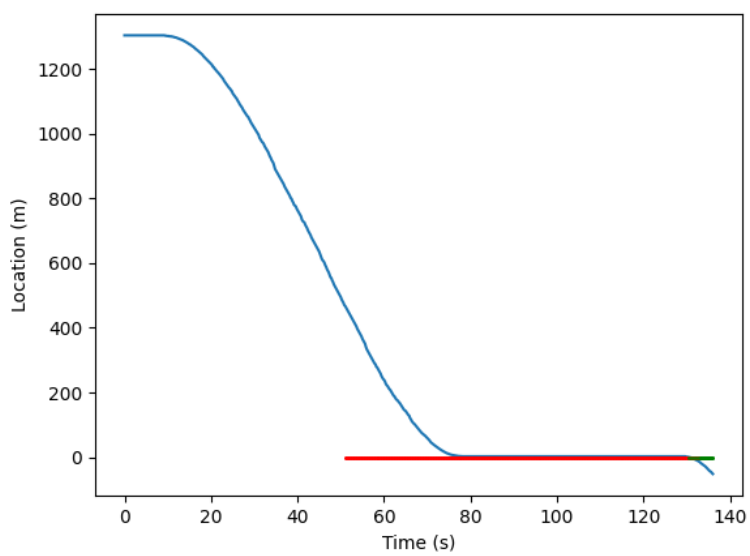


Figure 7.10: Target vehicle's longitudinal location along the road section. The signalized intersection's stop bar is at the location of 0m. The signal status of the time step when the target vehicle was in the RSU's communication range is also included.

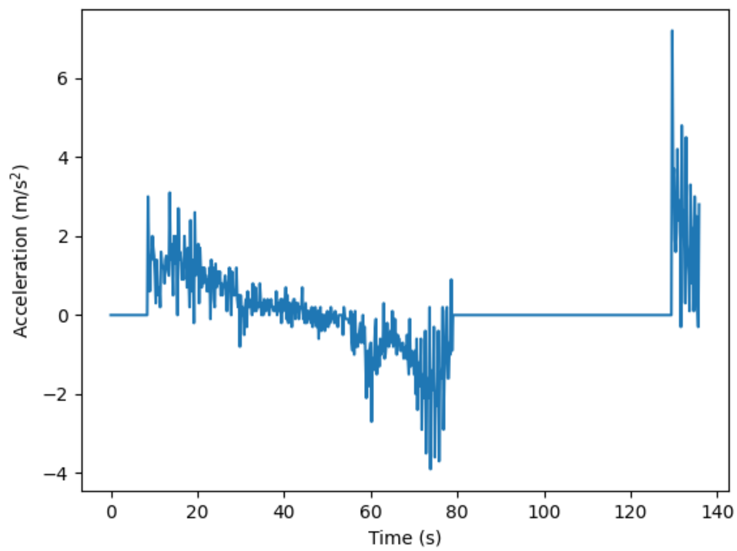


Figure 7.11: Target vehicle's acceleration when it drove towards the intersection during red light.

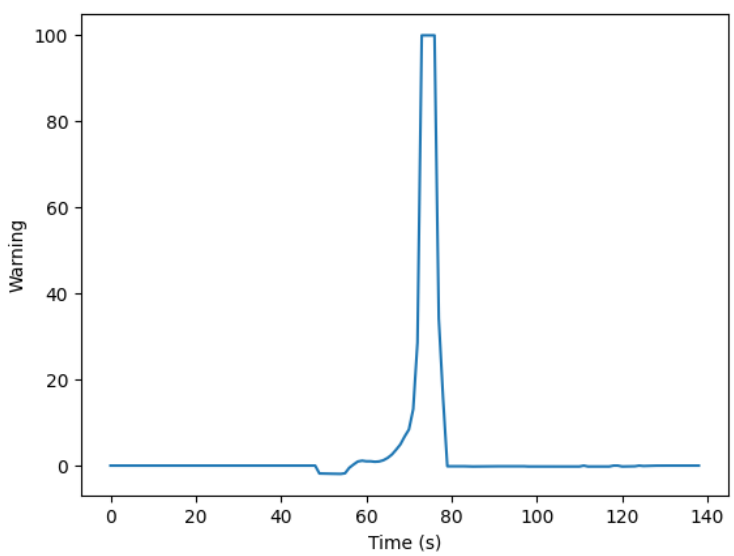


Figure 7.12: Target vehicle's warning magnitude when it drove towards the intersection during red light.

It is evident that the test vehicle decelerated and came to a stop (around 80s of the test drive) at the intersection since the traffic signal was red. It proceeded through the intersection once the light changed to green (around 130s of the test drive). The results show that the warning system has the ability to compute warning signal in real-time based on the information transmitted from the OBU. Notably, the warning system initiated the warning signal and advised the driver to reduce speed when the vehicle was about 300m away from the intersection (around 60s of the test drive). The vehicle's acceleration changed to negative after that. However, as the driver did not comply with the warning system's prompt to slow down, a larger warning signal was generated to ask the driver to employ a greater deceleration to avoid running a red light. It takes 5 steps (5 sec) for the warning signal to increase from 8.4 to its maximum value 100. Compared to the result in Task 4, the shorter communication range during the on-road test led to a delayed activation of the warning system, resulting in a steeper rise of the warning signal is observed.

7.4 Summary

In this study, a real-time implementable red light running warning system was developed and tested using the configured hardware-in-the-loop testbed. Traffic conditions including traffic signal status and the target vehicle's information were obtained through the OBU. The laptop was connected to the OBU and computed the optimal warning signal in real-time. It displayed warning message on the screen and could be observed by the researcher. The results showed that the proposed warning algorithm could be demonstrated in real-time. However, due to the utilization of actuated signal control technology at the intersection, further work is necessary to approximate the predicted traffic signal state along the prediction horizon. Further work will involve employing additional driving scenarios to further demonstrate the proposed algorithm in the upcoming months.

Chapter 8

On-road demonstration of RLRWS

May, 2024

8.1 Introduction

This project is focused on developing a technology to warn the driver regarding red light violations using vehicle-to-infrastructure (V2I) technology. In previous tasks, the team project proposed an algorithm that calculates deceleration suggestions for the target vehicle in real-time based on the current vehicle dynamics and traffic signal phasing. A microscopic traffic simulator was used to validate the proposed red-light running warning system. Subsequently, the presented warning algorithm is converted into a real-time implementable system. This task describes the RLRWS demonstration in the vehicle. Project team members have implemented the system and receive warning messages while driving the car. Setup instruments and muliple examples of in-car demonstration are presented in this report.

For real-time computation of the warning signal, the proposed system must have the capability to receive and interpret the connected vehicles messages, e.g. signal phase and timing (SPaT) from the roadside unit (RSU) and basic safety messages (BSM) from other connected vehicles in real-time. Then, a traffic prediction algorithm is employed to predict traffic conditions at each time step. Based on the output of the prediction algorithm, the optimization-based warning algorithm will compute the optimal warning signal in real-time. Finally, a visualization function will be included to display the warning message on the screen in real-time.

The research team installed the roadside unit (RSU) at the intersection of Scott County's CSAH 18/CSAH 21/Southbridge Parkway with the help of MnDOT in Task 1. Furthermore, an on-board unit (OBU) located in the car has access to vehicle dynamics (location and speed) and also receives SPaT from RSU. We validated the connectivity and reliability of the purchased RSU and OBU in Task 2. The microscopic traffic simulator and structure of the warning system are shown in Task 3 and Task 4, respectively. In Task 6, we implemented the real-time red light running warning system on a test vehicle. Finally, we will show how this system performs while driving the target vehicle towards Scott County's CSAH 18/CSAH 21/Southbridge Parkway intersection.

8.2 Real-time warning algorithm framework

8.2.1 Hardware equipment

In this subsection, the hardware used for this task along with their connection is introduced. To broadcast the SPaT information of the traffic signal, the RSU is connected to the traffic signal controller as shown in Fig. 8.1. The RSU broadcasts the SPaT message via C-V2X communication. The socket in the traffic signal cabinet is used to supply power to the RSU.

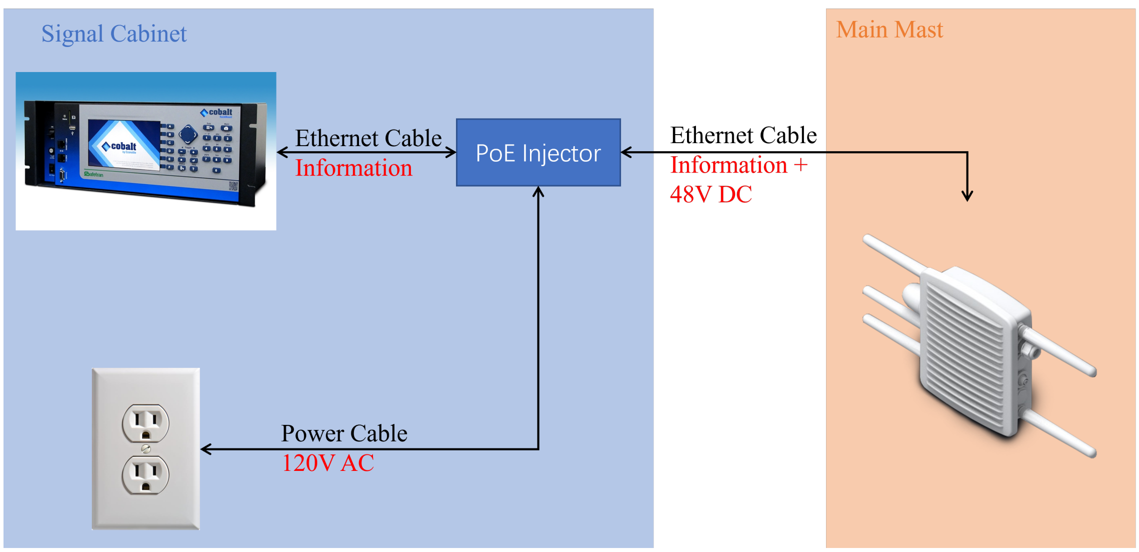


Figure 8.1: RSU configuration at the intersection.

The laptop and OBU are connected via an Ethernet cable in the test vehicles, where the OBU is powered by the vehicle's 12V DC power (see Fig. 8.2 and Fig. 8.3). The OBU receives the SPaT and BSM messages and sends them to the laptop in real-time via user datagram protocol (UDP). SPaT messages introduce the signalized intersection's traffic signal status. BSM messages contain the target vehicle's speed and location information. Utilizing the received data, scripts running on the laptop compute the optimal warning signal. Moreover, the laptop functions as the human-machine interface (HMI), displaying the warning signal on its screen. Ultimately, the driver can react to the warning message.

Figure 8.2: OBU and HMI configuration.

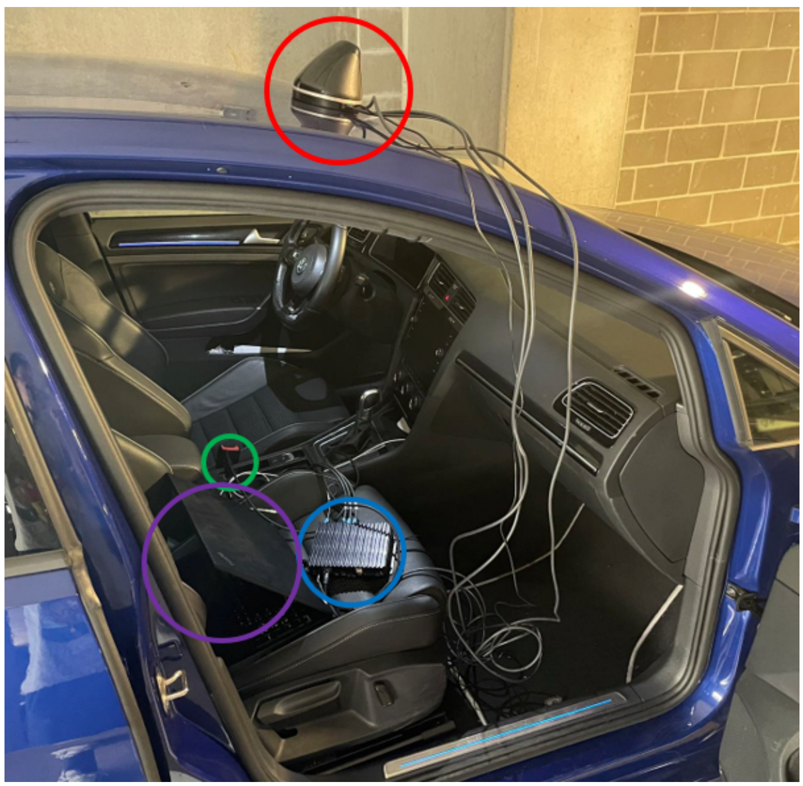


Figure 8.3: OBU installation for the test vehicle. The green circle indicates the vehicle 12V DC power supply. The red circle shows the antenna of the OBU. The blue circle shows the OBU processor. The purple circle represents the laptop used in this project.

8.2.2 Software framework

In this subsection, the software framework (Fig. 8.4) of the proposed real-time warning algorithm is presented. The developed framework employs Python as the programming language. To enhance computational speed, the entire warning algorithm framework is divided into four individual blocks, which will run in parallel:

- OBU information process

- Traffic prediction
- Warning algorithm
- Visualization

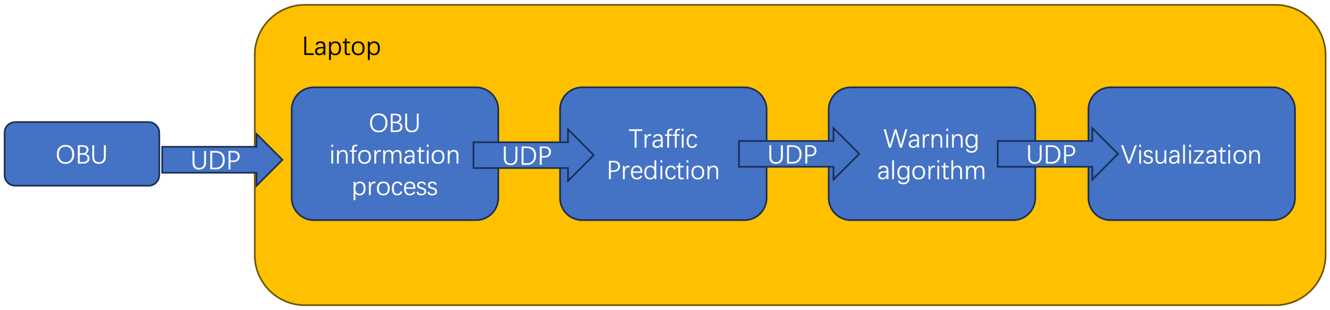


Figure 8.4: Software component of the real-time implementation of the proposed warning algorithm.

As demonstrated in the previous subsection, the UDP communication protocol facilitates real-time communication between the laptop and OBU. The UDP also serves as the communication standard among different blocks. UDP, known for its suitability in time-sensitive applications, enables faster communication since it does not spend time forming a firm connection with the destination before transferring the data. This attribute aligns well with the demands of developing this real-time warning algorithm.

For each programming block, a virtual buffer will be created to store received data from other programming blocks. These scripts will then read the most recent data from the buffer, ensuring access to the latest real-time system information. Further details about each programming block will be illustrated in the following subsections.

8.2.2.1 OBU information process block

This block receives and interprets the information transmitted from the OBU, and broadcasts the interpreted data to other programming blocks. To facilitate real-time broadcasting of the received message from the OBU to the laptop, the configuration file shown in Fig. 8.5 is stored in the device. The destination IP address and destination port number are specified in the configuration file. To receive the transmitted raw data stream in real-time, the "socket" package of Python [41] is used. The python package "pycmssdk" developed by Commsignia [30] is used to interpret the messages in real-time. Subsequently, these interpreted messages are broadcast to other programming blocks via UDP communication.

8.2.2.2 Traffic prediction block

This block utilizes real-time traffic information to predict traffic states along the prediction horizon. The traffic prediction algorithm developed in [39] without considering lane changing maneuvers is used in this project and its framework is shown in Fig. 8.6. In the figure, the red vehicle is the ego vehicle equipped with the proposed warning

```json
{
    "enabled": true,
    "filters": [
      {
        "direction": "In"
      },
      {
        "radioInterface": 1
      }
    ],
    "out": {
        "format": "Raw",
        "udp":{
          "destHost": "192.168.0.67",
          "destPort": 8000
        }
    },
    "source": [
      "Wsmp",
      "Gnp"
    ],
    "version": 6
  }
```

Figure 8.5: Configuration file in JSON format for real-time data transmission from Commsignia OBU.

algorithm. The road ahead of the ego vehicle is divided into M cells of equal length. V2V, V2I communication and perception technologies are used to obtain real-time traffic information, which is used by a state estimator to estimate the traffic conditions of each cell.

The project intersection utilizes an actuated signal timing plan that does not follow a fixed phase duration. It extends the phase duration according to vehicle actuation. It begins with a minimum value for the current phase; when a vehicle is actuated near the ending point of the phase, the phase receives an extension. If vehicles constantly approach the signal during this time, this process continues until it reaches a predefined maximum time. As a result, the ”min time to change” used in this task is subject to change.

To deal with the actuated signal used at the intersection, the variable “min time to change” is used in this task as it’s close to the actual signal timing during the last few seconds of each signal phase. The variable completely accounts for the actuated signal and will be increased in the case of an extension. Furthermore, even though the variable ”min time to change” is subject to change, the remaining time interval is sufficient for the RLRWS algorithm. The reason is that the only important phase in terms of red light running is when the green light turns yellow and then red. In this case, only the green phase is subject to extension and the yellow phase has a fixed time without extension. Therefore, the remaining time to provide a safe stop considering the yellow phase is sufficient for a target vehicle.

Moreover, in the case of approaching a red light, the extension is not against the RLRWS algorithm since the algorithm correctly suggests a safe stop. Finally, the future traffic states can be obtained by propagating the traffic flow model (8.1).

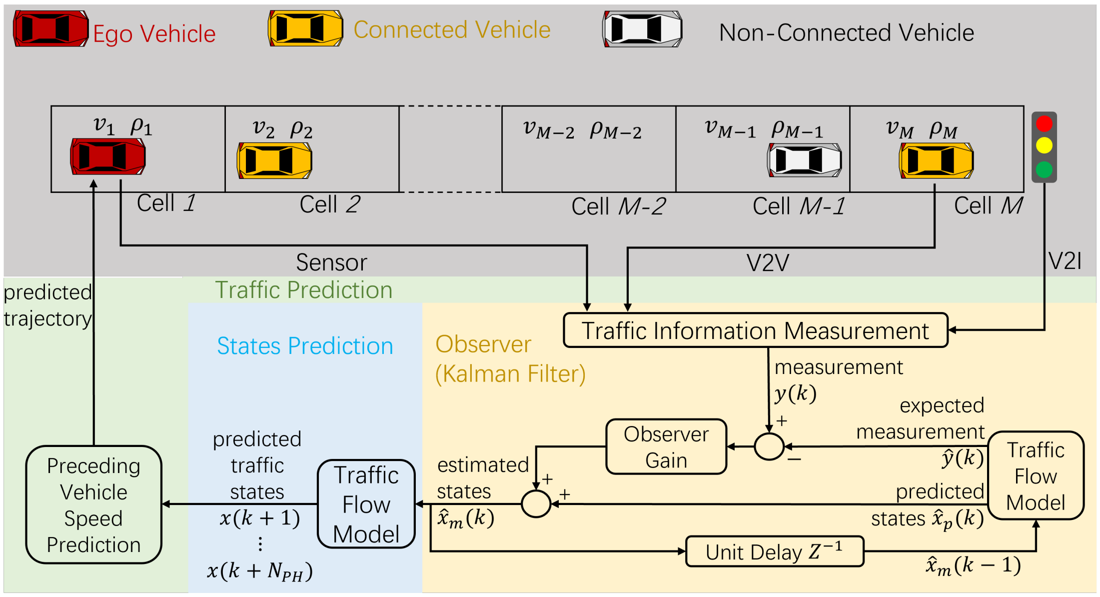


Figure 8.6: Traffic prediction framework used in the warning algorithm.

$$\rho_j(k+1) = \rho_j(k) - \frac{dt}{dx}\left[\rho_j(k)v_j(k) - \rho_{j-1}(k)v_{j-1}(k)\right] + \omega_j(k), \tag{8.1a}$$

$$v_j(k+1) = v_j(k) - \frac{dt}{dx}v_j(k)[v_j(k) - v_{j-1}(k)] + \underbrace{dt \cdot \frac{[\hat{V}_e(\rho_j(k)) - v_j(k)]}{\tau}}_{\text{Speed adaptation}}$$
$$\underbrace{-\frac{dt}{dx} \cdot \frac{c_0^2 \cdot [\rho_{j+1}(k) - \rho_j(k)]}{\rho_j(k) + \epsilon}}_{\text{Traffic pressure}} + \xi_j(k), \tag{8.1b}$$

where (8.1a) and (8.1b) describe the evolution of traffic density and traffic speed, respectively. k is the discretized time instance; dt denotes the time step size; dx represents the length of each cell; j is the cell index; ϵ is a small positive number to prevent zero denominator; $\omega_j(k)$ and $\xi_j(k)$ describe model uncertainties assumed to follow a Gaussian distribution; $\hat{V}_e(\rho_j(k))$ signifies the equilibrium speed of cell j; c_0 characterizes traffic pressure and τ describes the adaptation rate to reach the equilibrium speed.

Supposing an intersection is located at cell sc, then speed of this cell is set to zero when the traffic signal is red:

$$v_{sc}(k+1) = \begin{cases} 0, & \text{signal is red} \\ \text{right side of (8.1b)}, & \text{signal is green} \end{cases} \tag{8.2}$$

To facilitate real-time implementation, different from the work in [39], the discretization time dt has been adjusted from 0.1s to 0.2s. The prediction horizon is selected as 10s. The algorithm will update the traffic prediction output every 0.2s.

8.2.2.3 Warning algorithm block

This block uses real-time traffic information and the latest traffic prediction results to compute the optimal warning signal. As shown in the previous task, the proposed algorithm is formulated in the model predictive (MPC) fashion. To obtain the numerical solution for the formulated optimization problem, a time step with 0.2s is employed to discretize the problem. The predictive horizon is set to 10s. It is solved using IPOPT [42] with CasADi [43] used as modeling language. Considering that the computational time for solving the optimization problem is about 0.45s for each time step and humans cannot change behavior very often, the algorithm will update the warning message every 1s.

8.2.2.4 Visualization block

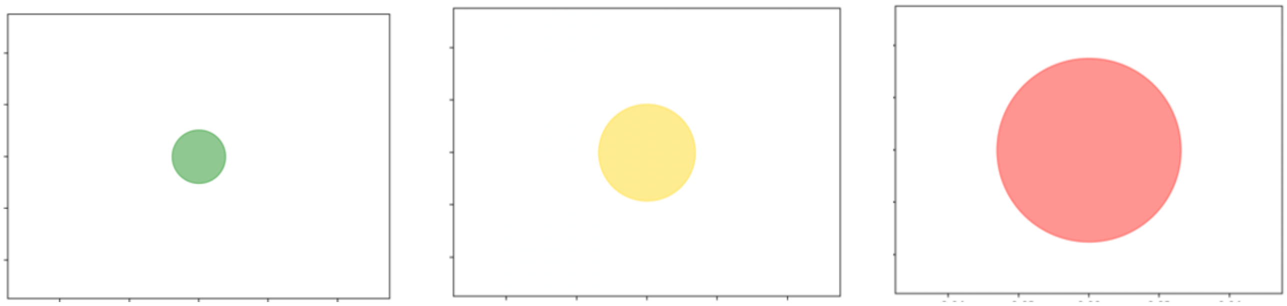
To visualize the warning message, the "matplotlib" package is employed to display the warning message through a figure on the laptop screen. As demonstrated in the previous task, three different colors (green, yellow, and red) along with varying sizes are used to display the warning signal. In this task, the warning message is displayed with colored circles as depicted in Fig. 8.7 (see Figs. 8.7a, 8.7b, and 8.7c) where the diameter change represents the required intensity, such as a large red circle meaning a harsh brake is suggested. A colored circle performs a faster visualization process compared to a more detailed horizontal percentage bar. The visualization block updates the output upon receiving the newest warning message.

Table 8.1 explains the color interpretation as intensity range. The green color represents normal driving behavior, and the RLRWS does not provide any additional guidance, encompassing all positive acceleration values as well as minor acceleration values. The yellow color indicates that the driver needs to make a moderate deceleration, including moderate deceleration rates (or moderate brake intensity rates), while the color switches to red when there is an imminent risk of a red-light violation with the necessity of an intense brake. Generally, the red color is displayed when the driver has not followed the yellow color bar displayed previously. As a result, yellow switches to red, and the brake intensity is increased in the displayed bar gradually as the driver did not follow the message.

Table 8.1: Warning message color classification

Warning Message	**Definition**
Green	No action required, far from signal or signal is green when the vehicle arrives
Yellow	Driver needs to decelerate gradually, a full stop is required soon
Red	A hard brake is required, otherwise will run the red light

An alternate approach to providing the warning is to release an alarm noise with an adjustable volume representing brake intensity. This approach is not implemented at this phase due to driver attention concerns such as confusion with another alarm sound (safety belt or other radars) or being distracted when listening to radio or music. Moreover, driver satisfaction might also drop when an alarm is released every time they approach a traffic signal even if they are not violating.

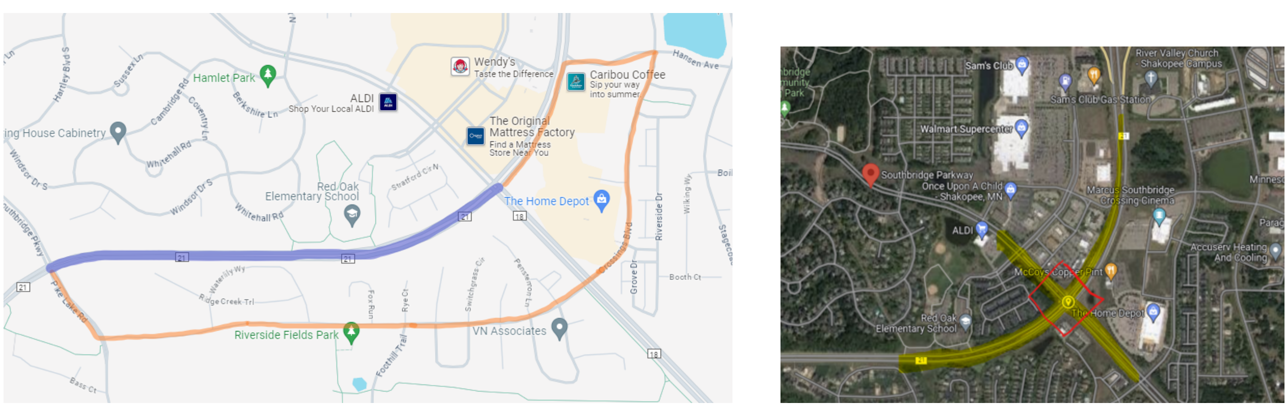


(a) Green warning output (b) Yellow warning output (c) Red warning output

Figure 8.7: Warning visualization cases

8.3 Results

Between December 2023 and May 2024, the research team went to Scott County's CSAH 18/CSAH 21/Southbridge Parkway intersection to test the proposed real-time warning framework. Fig. 8.8 is a map view of the project intersection and driving path for the target approach. As depicted in Fig. 8.8a, the Project team drove in CSAH 21 toward the intersection from West to North-East as the target approach is defined. The team tested the algorithm in real-time and recorded relevant data on the laptop for subsequent analysis if needed.



(a) Driving path for road test: blue line shows the main driving approach and orange line provides access to the target approach (b) Highlighted roads are locations that OBU can receive SPaT messages

Figure 8.8: Map views of Scott County's CSAH 18/CSAH 21/Southbridge Parkway intersection

During the test drive, the test vehicle traveled northbound on CSAH 21 toward the signalized intersection. Figure 8.9 shows the in-vehicle setup. OBU is utilized with three ports in this project: 1) a 12V power cable

Figure 8.9: Setup in the vehicle: the laptop is used to receive and decode BSM and SPaT messages, run the developed algorithm, and show the warning signal in real-time.

connected to the vehicle outlet 2) An antenna to receive SPaT and GPS data appropriately 3) An ethernet cable that outputs the required message to the laptop to calculate warning message. The computer runs RLRWS and presents the warning message as colored circles with varying sizes. A research team member in the passenger seat operates the laptop and communicates the warning message to the driver.

Figs. 8.10, 8.11, 8.12, and 8.13 present the warning message along with the vehicle's speed, and space-time plots in various cases. Results are recorded during the test drive where the test vehicle approached the intersection during a red phase. Signal phasing is displayed with colored dots in the space-time diagram in these figures. When the vehicle is far away from the intersection, SPaT messages are not transmitted to the car yet. Thus, there is no information available regarding the signal phase. When the target vehicle enters the connectivity range (500 m in this project), messages will be received. This is the reason that signal-phasing dots are not displayed at some points of the space-time graph.

It should be noted that there is a linear relation between the warning message and the suggested deceleration in terms of magnitude and mathematical sign with the acceleration formula $a = -\frac{\text{warning}}{20}$ (for example a warning value of 100 represents a deceleration of 5 meters per second square). Thus, a positive warning value corresponds to brake guidance, and a negative warning value guides acceleration.

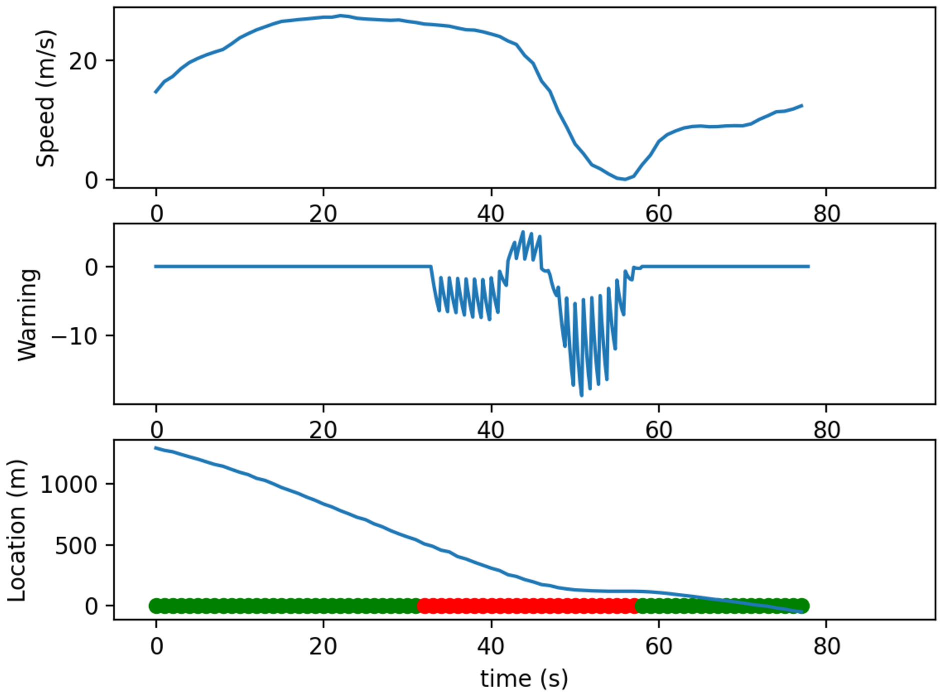


Figure 8.10: speed, warning value, and space-time plots for a recorded road test

Fig. 8.10 represents a case when the target vehicle approaches the intersection during the red phase. As the driver slows down earlier than what is calculated by RLRWS (around the time 25s), the algorithm guides the driver to accelerate (negative warning value). This suggestion is interpreted as normal driving since it does not result in a red light violation. Thus, RLRWS displays a fixed-size green circle (Fig. 8.7a). Subsequently, the algorithm begins deceleration suggestion at a time around 42–48 seconds with a yellow varied-size circle (Fig. 8.7b).

Fig. 8.11 depicts a harsh brake suggestion. It begins with suggesting normal driving behavior. As braking is required, the algorithm starts showing a yellow color (around the time 28 s) followed by a harsh brake suggestion with a red-colored circle. It is evident that the test vehicle decelerated and came to a stop (around 38s of the test drive) at the intersection since the traffic signal was red. It proceeded through the intersection once the light changed to green (around 63s of the test drive).

Fig. 8.12 presents another experimental test with the same driving setting while approaching a red light. Since the driver tends to follow RLRWS guidance, the warning message does not switch to red color, unlike the previously mentioned case. RLRWS suggests mild braking by displaying a yellow color warning message with a moderate intensity.

As the last example, Fig. 8.13 shows a scenario when the traffic signal is initially green but the driver does

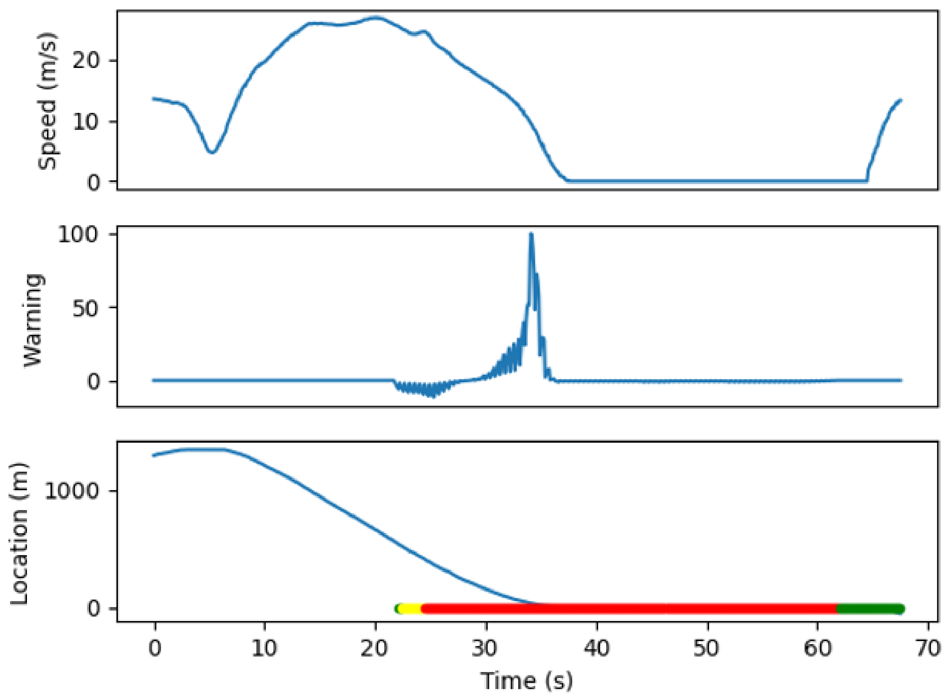


Figure 8.11: speed, warning value, and space-time plots for a recorded road test

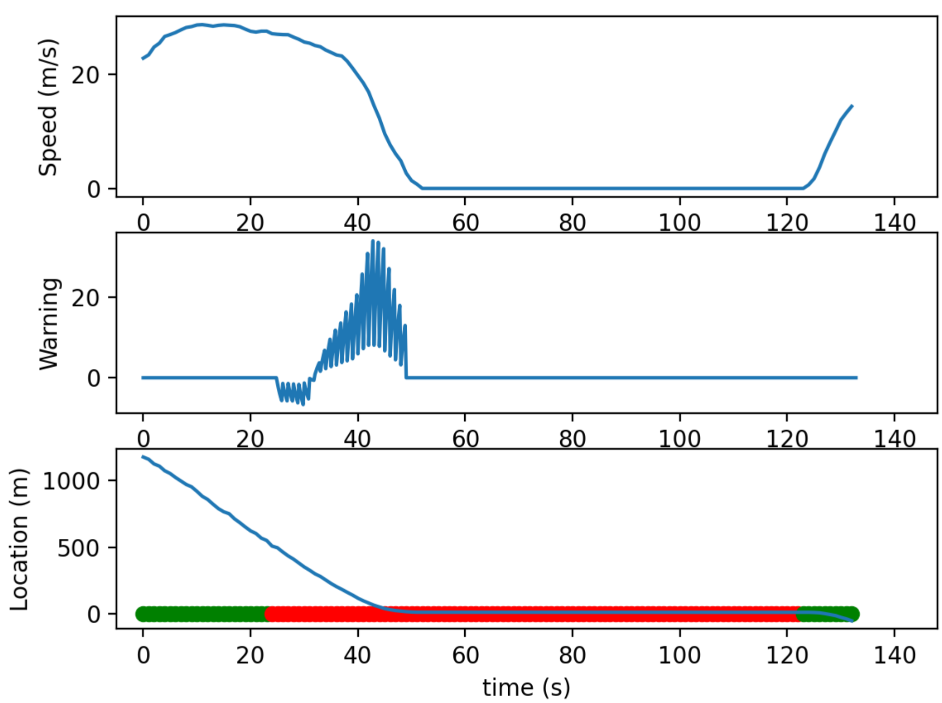


Figure 8.12: speed, warning value, and space-time plots for a recorded road test

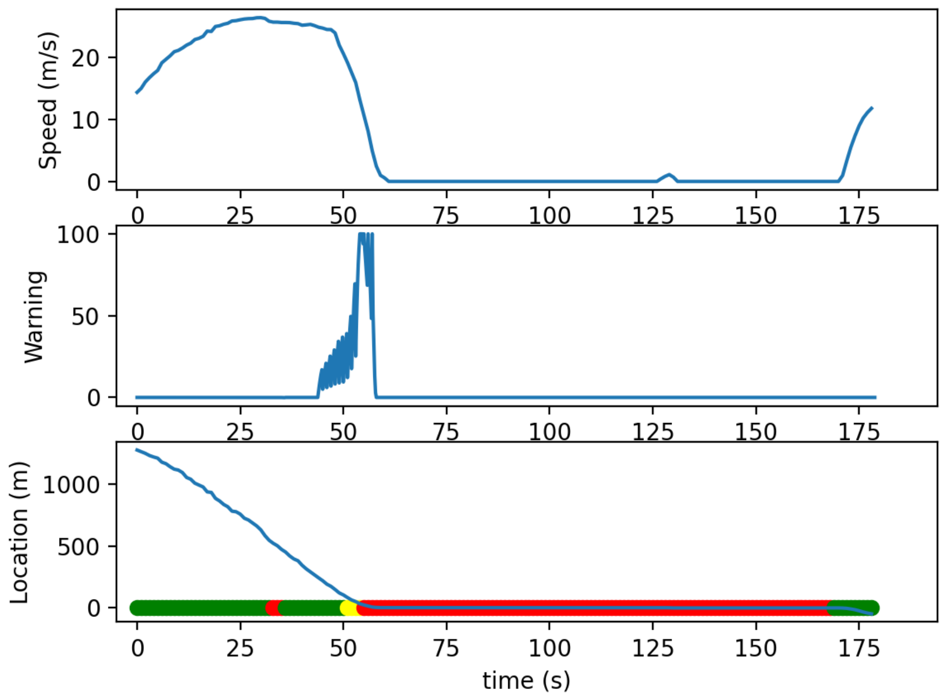


Figure 8.13: speed, warning value, and space-time plots for a recorded road test

not follow RLRWS suggestion (mild braking) and keeps driving at the free flow speed. As the driver approaches the intersection, it switches to yellow and then red (around time 50 to 55 seconds). It can be noted that a mild braking suggestion begins even when the light is in the green phase since RLRWS predicts that the light will turn red before the driver can cross through the intersection. As the driver keeps going fast and ignores the yellow-colored warning message, RLRWS starts to warn the driver with a harsh deceleration suggestion displayed with the red light (around time 51 to 57 seconds).

The results show that the warning system has the ability to compute warning signal in real-time based on the information transmitted from the OBU. Notably, the warning system initiated the warning signal and advised the driver to reduce speed when the vehicle was about 300m away from the intersection.

8.4 Summary

In this study, a real-time red light running warning system was developed and demonstrated using the configured hardware-in-the-loop testbed. Traffic conditions including traffic signal status and the target vehicle's information were obtained through the OBU. The laptop was connected to the OBU and computed the optimal warning signal in real-time. It displayed a warning message on the screen that successfully guides the driver in various cases. The results showed that the proposed warning algorithm can visualize and warn the driver depending on vehicle dynamics and signal phase. Finally, a connection between vehicles and intersections followed by employing RLRWS can prevent red light violations which is a major source of fatalities and injuries.

Chapter 9

Memorandum on expected research benefits and potential implementation steps

9.1 Introduction

Safety is critical for on-road transportation. Among all factors that can result in serious accidents, red light running happens frequently and is often deadly. According to statistics from the Insurance Institute for Highway Safety (IIHS) [1], 928 people were killed in crashes that involved red light running in 2020 within the United States. Additionally, an estimated 116,000 people were injured in accidents related to red light running. It is obvious that developing effective countermeasures for red light running violations can significantly reduce the number of these accidents and improve road safety. One solution is to develop a red light running warning system (RLRWS), which can provide a warning signal to drivers about the possibility of running a red light in advance.

RLRWS is heavily based on vehicle connectivity; equipped with dedicated short range communication (DSRC) or cellular vehicle-to-everything (C-V2X) devices, CVs can obtain real-time traffic information such as Signal Phase and Timing (SPaT), as well as speed and location data from other CVs, via Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V) communication, respectively. The obtained traffic information can be used to estimate the current traffic state ahead of the vehicle and predict future traffic traffic states. The vehicle's future longitudinal position can then be computed based on the predicted traffic states. Therefore, it is necessary to install communication devices to enable the V2I and V2V communication for this research project. Equipment and installation form the major part of the costs. Specifically, the roadside unit (RSU) and onboard unit (OBU) are required.

This report presents a comprehensive analysis of the Red Light Running Warning System (RLRWS) deployment. It begins by detailing the costs associated with implementation, including equipment procurement, human resource requirements for installation, and framework development expenses. The report then explores the potential benefits of the system, which are categorized into three main areas: cost-effectiveness in crash reduction, road maintenance savings, and improvements in traffic flow. Both costs and benefits are categorized and quantified for a practical network size, providing a realistic assessment of the RLRWS's economic and operational impact. This approach offers a balanced view of the system's feasibility and potential return on investment for urban planners and transportation authorities.

9.2 Cost breakdown

The implementation of the Red Light Running Warning System (RLRWS) requires specific hardware components, which constitute a significant portion of the project's costs. The system's core functionality relies on a two-way communication setup:

1. Intersection Equipment: Each signalized intersection must be equipped with a Roadside Unit (RSU). The RSU is responsible for transmitting real-time signal information.

2. Vehicle Equipment: Target vehicles need to be fitted with an On-Board Unit (OBU). The OBU serves two crucial functions: a) Receiving data from the RSU, b) Providing vehicle dynamics (such as speed, GPS coordinates, and acceleration) as essential inputs for the RLRWS algorithm

To ensure seamless communication and compatibility, both RSUs and OBUs should support standard V2X messaging formats defined by the Society of Automotive Engineers (SAE). from the same manufacturer. For this project, we have opted to purchase both units from the same manufacturer, which might affect the overall equipment costs but guarantees system integrity as a demonstration project.

The RSU device works closely with the signal controller, which is often located in a cabinet near the intersection. Therefore, only authorized experts can mount and install the RSU, which requires additional effort and costs. However, OBU devices can be simply installed in a car with a power source (e.g., a 12-volt outlet available inside the vehicle).

The red light violation is studied separately for each intersection in this project. Therefore, we report the cost of RSU installation for each individual intersection. Each RSU is able to communicate with multiple

Table 9.1: RLRWS implementation costs in USD

item	Unit price USD
RSU kit	$ 3020
OBU kit	$ 350
configuration, software, and licensee	$ 4850
workforce per RSU installation	$ 850

approaching vehicles equipped with OBUs. RSU and OBU devices used in this project were loaded with various features and processing power in order to test other applications of SPaT messages in the future. Thus, the OBU kit was priced at $2040 which is higher than a less-featured OBU with V2V compatibility which might be sufficient for the RLRWS purpose. For mass-market OBUs, the National Highway Traffic Safety Administration (NHTSA) has estimated a unit cost of $350 per vehicle in 2020 [44]. Table 9.1 describes the unit costs to purchase and install RSUs and OBUs. It should be noted that each intersection only requires one RSU plus installation, each vehicle requires an OBU, and we assume that the software and licensing costs are one-time only.

The total number of registered automobiles in Minnesota in 2020 is reported as around 1,827,000 vehicles [45]. The project team proposes targeting 10% of more vulnerable vehicles for RLRWS implementation which is around 182,700 vehicles (e.g. demographic and geographic filters) with the total OBU purchase cost of:

$$182700 \times \$350 = \$63,945,000 \tag{9.1}$$

Assuming a reasonable radius of the Twin Cities metro area (radius of 50 miles) to run RLRWS and install RSUs which attract most of the traffic volume (For example, Ramsey and Hennepin counties located in Twin Cities account for about 30% of fatal crashes in Minnesota). This area approximately has 640 signalized intersections according to the Metro District Traffic Engineering webpage (MnDOT). Therefore, RSU installation costs for this network is as follows:

$$\$4850 + (\$850 + \$3020) \times 640 = \$2,481,650 \tag{9.2}$$

It can be noted that the major costs are related to vehicle connectivity technology (OBU device). Finally, the total costs for equipping 10% of vehicles while driving in a relatively large area around the Twin Cities will be as follows:

$$\$63,945,000 + \$2,481,650 = \$66,426,650 \tag{9.3}$$

Installing OBUs and RSUs, even on a portion of intersections and vehicles, creates a connectivity infrastructure that enables numerous advantages beyond RLRWS. Energy-efficient approaches (referred to as eco-approach) can optimize vehicle speeds, reducing fuel consumption and emissions [46, 18]. Safety improvements using vehicle-to-vehicle (V2V) information transmission allow for real-time hazard warnings and cooperative adaptive cruise control [47]. Furthermore, it lays the groundwork for future autonomous vehicle integration and enhanced pedestrian safety through vehicle-to-pedestrian (V2P) communication [47]. While the installation costs are required for the RLRWS project, equipping roads and vehicles with this technology creates a versatile, future-proof framework. This investment opens doors to a comprehensive set of opportunities and benefits, positioning it as a strategic step towards a more intelligent, efficient, and safer transportation ecosystem.

9.3 Potential benfits

The implementation of RLRWS offers a range of potential benefits that extend beyond its primary function of reducing crashes due to traffic violations. This section of the report explores these advantages, categorizing them into three main areas: cost-effectiveness in crash reduction, road maintenance savings, and improvements in traffic flow. Each of these categories represents a significant potential for a positive impact on both public safety and urban infrastructure management.

1. Cost-effectiveness in crash reduction and citation avoidance: The RLRWS aims to significantly reduce the incidence of red light violations, which are a leading cause of intersection crashes. By providing timely warnings to drivers, the system has the potential to prevent accidents, thereby saving lives and reducing the economic burden associated with traffic collisions. Additionally, by helping drivers avoid running red lights, the system will lead to fewer traffic citations, resulting in direct financial savings for drivers. This section will quantify the potential reduction in crash rates, estimate the resulting economic savings in terms of reduced medical costs, property damage, and lost productivity, and calculate the potential savings for drivers from avoided traffic citations.

2. Road maintenance savings: A less obvious but important benefit of the RLRWS is its potential to reduce wear and tear on road infrastructure. By promoting smoother traffic flow and reducing the frequency of sudden stops and starts at intersections, the system may contribute to decreased road surface degradation [48]. One benefit of RLRWS is the potential long-term savings in road maintenance costs, including reduced frequency of resurfacing and repair work at intersections. Moreover, crashes caused by red light violations can heavily damage road pavement and surroundings (e.g. shoulders and guardrails); avoiding those damages utilizing RLRWS can reduce road maintenance significantly.

3. Improvements in traffic flow: The RLRWS has the potential to optimize traffic flow by reducing the number of red light violations and the resulting traffic disruptions. This improvement can lead to reduced travel times, lower fuel consumption, and decreased emissions from idling vehicles. The section will analyze the potential gains in traffic efficiency, estimating reductions in average travel time through equipped intersections and the

associated economic and environmental benefits.

Given that the impact and magnitude of crash reduction are more significant than other benefit categories, we aim to quantify this item precisely and estimate other elements as a proportion of this cost. It's important to note that various costs associated with accidents, particularly those involving fatalities and major injuries, are critically important to avoid utilizing RLRWS. However, for the purposes of this report, we can only express the economic benefits of the RLRWS in U.S. dollars.

There are various causes of collisions, such as speeding, alcohol-related incidents, yield and merge failures, and running red lights. The Minnesota Department of Public Safety has categorized these reasons and reported the share of each cause in the total number of crashes per year [49]. The project team has identified several causes for crashes that can be avoided by utilizing RLRWS:

- Red light running: This cause can be completely addressed by RLRWS as it warns drivers when running a red light.

- Speeding: There are various cases in which drivers might violate the speed limit. Among those cases, speeding might occur while approaching a connected intersection. Based on the real-time signal status, RLRWS can warn drivers to either slow down for a red light or follow the speed limit for the green light. Thus, we can assume that RLRWS suggestions can reduce speeding-related crashes by at least 20%.

- Careless driving, distraction, and disregarding traffic signals: As RLRWS displays real-time warning messages near intersections, drivers are expected to pay more attention to their driving and current speed, particularly when a red-colored message is displayed. As a result, the project team expects these two crash causes to be reduced by 20%.

- Improper merging and turning: RLRWS can help warn drivers even if they are planning to make a right or left turn at the intersection. The reason is that RLRWS guides the drivers to provide a safe stop before any turn. Although it does not cover all intersections and yields, it can address a portion of crashes related to improper merging and turning. We expect this cause to be reduced to half by using RLRWS (address turn failure causes).

In 2023, Minnesota experienced 70,266 traffic crashes, resulting in 444 fatalities and 23,704 injuries [49]. These crashes resulted in a total economic loss of over $2,239,000,000. Among the behavioral contributions resulting in a crash, some dangerous behaviors can be avoided or reduced by utilizing RLRWS as mentioned, including running the red light with a 4% contribution that RLRWS can completely address, disregarding traffic signals with 2% contribution reduced to 1% (50% reduction), improper merge and turn with 5% contribution reduced to 3% (50% reduction), speeding with 18% contribution reduced to 16% (20% reduction), driver distraction with 7% contribution reduced to 6% (20% reduction), and careless driving with 8% contribution reduced to 7% (20% reduction). Although the crash reduction values are assumptions made by authors, adding up those reductions will show that at least 11% of them can be prevented by following RLRWS optimistically (e.g. running a red light, turn and merge failures, driver distraction, etc.) which can potentially result in above 7,700 crash reductions.

The RLRWS demonstration in a large radius of Twin Cities can demographically cover 65% of crashes within the state of Minnesota. This number is obtained by comparing the Twin Cities metro area's population (3.7 million) to the state of Minnesota's population (5.7 million) [50] in 2022. Overall, the potential benefit in terms of crash reduction is as follows (10% loss reduction):

$$\$2,239,000,000 \times 0.11 \times 0.65 = \$160,088,500 \tag{9.4}$$

All other potential benefits such as traffic citation, maintenance, and traffic improvement can be added as an additional 15 % on top of that making the total benefit per year as follows:

$$\$160,088,500 \times 1.15 = \$184,101,775 \tag{9.5}$$

9.4 Conclusions

The implementation of the Red Light Running Warning System (RLRWS) in the Twin Cities metro area presents a compelling case for improving traffic safety and efficiency. Our analysis reveals that the initial costs for a limited implementation, covering 640 signalized intersections and equipping 10% of vehicles in Minnesota, amount to $66,426,650. This investment, while substantial, is projected to yield significant returns.

The annual benefits of the RLRWS are estimated at $184,101,775, encompassing reduced crash-related costs, fewer traffic citations, road maintenance savings, and improved traffic flow. Notably, these benefits are expected to be observable even within the first year of implementation, suggesting a rapid return on investment.

The aggregate benefit to the economy is clear and substantial. The RLRWS has the potential to prevent approximately 7,700 crashes annually in the target area, leading to significant savings in terms of medical costs, property damage, and lost productivity. Moreover, the system's impact extends beyond direct crash prevention, contributing to smoother traffic flow and reduced infrastructure wear.

However, it is important to note that while the aggregate benefits are substantial, the impact may not be immediately apparent to every individual driver. Many drivers may not experience a dangerous situation that would have resulted in a crash without the RLRWS. The system's value lies in its preventive nature, averting potential accidents before they occur.

Despite this potential disparity in perceived individual benefit, the overall societal and economic advantages of the RLRWS are clear. The system represents a proactive approach to traffic safety, aligning with broader goals of reducing traffic fatalities and injuries. As connected vehicle technology becomes more widespread, the benefits of systems like the RLRWS are likely to increase, potentially leading to even greater returns on investment in the future.

In conclusion, the RLRWS demonstrates a promising approach to enhancing road safety and efficiency. While the initial investment is significant, the projected benefits, both in terms of lives saved and economic impact, make a strong case for its implementation. As we move forward, continued monitoring and evaluation of the system's performance will be crucial to optimize its effectiveness and ensure it delivers on its potential to create safer, more efficient roads for all users.

Chapter 10

Conclusions

The scope of this research are to develop a red-light running warning system that reduces red-light running violations and improves safety of signalized intersections. Signal phasing and timing data along with a vehicle's longitudinal movement information were used to predict the target vehicle's future movements. Then a model-predictive control-based algorithm was deployed to compute the optimal warning signal, which was then shown to the driver on the laptop screen. Simulated traffic scenarios were used to show the performance of the proposed framework. Furthermore, an on-road testbed containing RSU and OBU was built and was employed for experimental validation of the algorithm.

The overall cost of deploying such a warning system us based on RSU, OBU equipment, installations, and support. The overall benefit is based on the reduction of crash-relevant costs, fewer traffic citations, road maintenance savings, and improved traffic flow. These benefits are expected to be seen in the first year of implementation, which means the deployment of the system will give a rapid return. Based on our estimation, when we consider the whole Twin Cities metro area, covering 640 signalized intersections and equipping 10% of vehicles in Minnesota with the system, the cost amounts to $66,426,650. The annual benefits of the RLRWS are estimated at $77,245,500 and it will also prevent approximately 7,000 crashes annually in the target area.

However, there remains a significant gap between the current algorithm and a market-ready, implementable software. The current algorithm uses a simplified linear driver model to describe the driver's response to the warning signal. However, driver reactions are influenced by a range of factors, including individual driving behavior, road conditions, weather, and other variables. Therefore, future efforts should focus on creating a more comprehensive driver model that accurately captures these dynamics. Meanwhile, the inherent uncertainty in driver behavior should also be considered in the algorithm design, as human drivers may not always follow the guidance of the system perfectly. Further test drives involving multiple connected vehicles, various signalized intersections and different drivers should also be conducted in the future to validate the effectiveness of the proposed system across different scenarios and drivers.

Bibliography

[1] IIHS, "Red Light Running." Retrieved from https://www.iihs.org/topics/red-light-running#:~:text=Overview,to%20discourage%20red%20light%20running, 2024.

[2] N. Elmitiny, E. Radwan, X. Yan, and M. Jardaneh, "Field testing of a proposed pavement marking as a red light running countermeasure," *The Open Transportation Journal*, vol. 4, no. 1, 2010.

[3] X. Yan, E. Radwan, D. Guo, and S. Richards, "Impact of 'signal ahead' pavement marking on driver behavior at signalized intersections," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 12, no. 1, pp. 50–67, 2009.

[4] C. Goldenbeld, S. Daniels, and G. Schermers, "Red light cameras revisited: Recent evidence on red light camera safety effects," *Accident Analysis & Prevention*, vol. 128, pp. 139–147, 2019.

[5] E. G. Cohn, S. Kakar, C. Perkins, R. Steinbach, and P. Edwards, "Red light camera interventions for reducing traffic violations and traffic crashes: A systematic review," *Campbell Systematic Reviews*, vol. 16, no. 2, p. e1091, 2020.

[6] W. Hu and J. B. Cicchino, "Effects of turning on and off red light cameras on fatal crashes in large us cities," *Journal of Safety Research*, vol. 61, pp. 141–148, 2017.

[7] K. P. Shannon, "Speeding towards disaster: How cleveland's traffic cameras violate the ohio constitution," *Clev. St. L. Rev.*, vol. 55, p. 607, 2007.

[8] E. Polders, J. Cornu, T. De Ceunynck, S. Daniels, K. Brijs, T. Brijs, E. Hermans, and G. Wets, "Drivers' behavioral responses to combined speed and red light cameras," *Accident Analysis & Prevention*, vol. 81, pp. 153–166, 2015.

[9] Q. Hussain, W. K. Alhajyaseen, K. Brijs, A. Pirdavani, and T. Brijs, "Innovative countermeasures for red light running prevention at signalized intersections: A driving simulator study," *Accident Analysis & Prevention*, vol. 134, p. 105349, 2020.

[10] H. Köll, M. Bader, and K. W. Axhausen, "Driver behaviour during flashing green before amber: A comparative study," *Accident Analysis & Prevention*, vol. 36, no. 2, pp. 273–280, 2004.

[11] T. J. Smith and R. Harney, "Effects of advanced warning flashers on driver interaction with signalized intersections," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 45, pp. 1680–1684, SAGE Publications Sage CA: Los Angeles, CA, 2001.

[12] K. Long, Y. Liu, and L. D. Han, "Impact of countdown timer on driving maneuvers after the yellow onset at signalized intersections: An empirical study in changsha, china," *Safety Science*, vol. 54, pp. 8–16, 2013.

[13] K. Long, L. D. Han, and Q. Yang, "Effects of countdown timers on driver behavior after the yellow onset at chinese intersections," *Traffic Injury Prevention*, vol. 12, no. 5, pp. 538–544, 2011.

[14] C. Fu, Y. Zhang, W. Qi, and S. Cheng, "Effects of digital countdown timer on intersection safety and efficiency: a systematic review," *Traffic Injury Prevention*, vol. 17, no. 1, pp. 98–103, 2016.

[15] X. Yan, Q. Xue, L. Ma, and Y. Xu, "Driving-simulator-based test on the effectiveness of auditory red-light running vehicle warning system based on time-to-collision sensor," *Sensors*, vol. 14, no. 2, pp. 3631–3651, 2014.

[16] S. Elias, M. Ghafurian, and S. Samuel, "Effectiveness of red-light running countermeasures: A systematic review," in *Proceedings of the 11th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pp. 91–100, 2019.

[17] Y. Zhang, X. Yan, and X. Li, "Effect of warning message on driver's stop/go decision and red-light-running behaviors under fog condition," *Accident Analysis & Prevention*, vol. 150, p. 105906, 2021.

[18] Y. Shao and Z. Sun, "Energy-efficient connected and automated vehicles: Real-time traffic prediction-enabled co-optimization of vehicle motion and powertrain operation," *IEEE Vehicular Technology Magazine*, vol. 16, no. 3, pp. 47–56, 2021.

[19] V. C. T. Committee, *V2X Communications Message Set Dictionary J2735_202007*, July 2020.

[20] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE internet of Things Journal*, vol. 1, no. 4, pp. 289–299, 2014.

[21] J. B. Kenney, "Dedicated short-range communications (dsrc) standards in the united states," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[22] I. C. S. L. S. Committee, "Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11*, 2007.

[23] A. Papathanassiou and A. Khoryaev, "Cellular v2x as the essential enabler of superior global connected transportation services," *IEEE 5G Tech Focus*, vol. 1, no. 2, pp. 1–2, 2017.

[24] M. of Transportation and Infrastructure, "Programming guide econolite cobal controller," 2019.

[25] Commsignia, "High performance v2x enabled roadside unit with edge computing." Retrieved from `https://www.commsignia.com/products/rsu/`, 2024.

[26] Commsignia, "Powerful v2x onboard unit." Retrieved from `https://www.commsignia.com/products/obu/`, 2024.

[27] DOT, "Isd message creator." Retrieved from `https://webapp2.connectedvcs.com/isd`, 2024.

[28] Marben, "Marben asn.1 solutions: Free online automotive asn.1 messages decoder." Retrieved from `https://www.marben-products.com/decoder-asn1-automotive/`, 2024.

[29] N. Probert, "Wireshark-dsrc, github repository." Retrieved from `https://github.com/nprobert/Wireshark-DSRC`, 2024.

[30] Commsignia, "V2x software stack." Retrieved from `https://www.commsignia.com/products/v2x-software-stack/`, 2024.

[31] F. H. Administration, "Adaptive signal control technology." Retrieved from `https://www.fhwa.dot.gov/innovation/everydaycounts/edc-1/asct.cfm`, 2024.

[32] X. Huang, D. Zhao, and H. Peng, "Empirical study of dsrc performance based on safety pilot model deployment data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2619–2628, 2017.

[33] DLR, "Emergency Vehicle Simulation," techreport, Simulation of Urban MObility (SUMO), 2024.

[34] C. Y. D. Yang and W. G. Najm, "Analysis of red light violation data collected from intersections equipped with red light photo enforcement cameras," techreport, Volpe National Transportation Systems Center, 2006.

[35] DLR, "speed mode," techreport, Simulation of Urban MObility (SUMO), 2024.

[36] DLR, "Traffic Control Interface," techreport, Simulation of Urban MObility (SUMO), 2024.

[37] M. W. Levin, Z. Sun, S. Wang, W. Sun, S. He, B. Suh, G. Zhao, J. Margolis, and M. Zamanpour, "Cost/benefit analysis of fuel-efficient speed control using signal phasing and timing (spat) data: Evaluation for future connected corridor deployment," Minnesota Department of Transportation, 2023.

[38] K. C. Dey, A. Rayamajhi, M. Chowdhury, P. Bhavsar, and J. Martin, "Vehicle-to-vehicle (v2v) and vehicle-to-infrastructure (v2i) communication in a heterogeneous wireless network–performance evaluation," *Transportation Research Part C: Emerging Technologies*, vol. 68, pp. 168–184, 2016.

[39] S. He, S. Wang, Y. Shao, Z. Sun, and M. W. Levin, "Real-time traffic prediction considering lane changing maneuvers with application to eco-driving control of electric vehicles," in *2023 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1–7, IEEE, 2023.

[40] W. Sun, S. Wang, Y. Shao, Z. Sun, and M. W. Levin, "Energy and mobility impacts of connected autonomous vehicles with co-optimization of speed and powertrain on mixed vehicle platoons," *Transportation Research Part C: Emerging Technologies*, vol. 142, p. 103764, 2022.

[41] G. M. Hassan, N. M. Hussien, and Y. M. Mohialden, "Python tcp/ip libraries: A review," *International Journal Papier Advance and Scientific Review*, vol. 4, no. 2, pp. 10–15, 2023.

[42] L. T. Biegler and V. M. Zavala, "Large-scale nonlinear programming using ipopt: An integrating framework for enterprise-wide dynamic optimization," *Computers & Chemical Engineering*, vol. 33, no. 3, pp. 575–582, 2009.

[43] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.

[44] M. Steadman and B. Huntsman, "Connected vehicle infrastructure: Deployment and funding overview," Texas A&M Transportation Institute, 2018.

[45] Office of highway policy information, "State motor-vehicle registrations," Federal Highway Administration, 2020.

[46] M. Zamanpour, S. He, M. W. Levin, and Z. Sun, "Incorporating lane-change prediction into energy-efficient speed control of connected autonomous vehicles at intersections," *arXiv preprint arXiv:2407.15004*, 2024.

[47] S. Biswas, R. Tatchikou, and F. Dion, "Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety," *IEEE Communications Magazine*, vol. 44, no. 1, pp. 74–82, 2006.

[48] H. Wang and I. L. Al-Qadi, "Evaluation of surface-related pavement damage due to tire braking," *Road Materials and Pavement Design*, vol. 11, no. 1, pp. 101–121, 2010.

[49] Minnesota Department of Public Safety, "Minnesota motor vehicle crash facts," 2020. Saint Paul, MN: Alternative Fuels Data Center.

[50] United Stated Census Bureau, "2022 minnesota populations and people," 2022. Saint Paul, MN: Alternative Fuels Data Center.