Cover Sheet

Submitted to	DOE, EERE, project managed by National Energy Technology Laboratory
FOA-0001213	FY2015 VEHICLE TECHNOLOGIES OFFICE INCUBATOR
Nature of Report	Final Report
Award number	DE-EE0007212
Award type	Grant
Primary Award institution	University of Michigan
PI	Prof. Huei Peng hpeng@umich.edu 734-769-6553
Prime Recipient type	University
Project title	Energy Impact of Connected and Automated Vehicle Technologies
DUNS number	073133571
Date of report	March 30, 2020
Period covered by the report	October 1, 2015 - November 30, 2019

Acknowledgment: "This material is based upon work supported by the Department of Energy, Office of Energy Efficiency and Renewable Energy (EERE), under Award Number DE-EE0007212."

Executive Summary

The overarching goal of this project is to understand the potential impact of connected and automated vehicles. The goal was achieved through data collection, model development, algorithm designs, simulations, and limited field tests.

The main outcomes from this project include: (1) we collected energy consumption and GPS data from 500 vehicles over one year, with a total mileage of 8 million miles; (2) Based on the collected data and other datasets collected at the University of Michigan, we developed a calibrated Ann Arbor model in Polaris (model developed by ANL), and the fuel economy accuracy was found to be around 3.9% by comparison with field collected data; (3) An open-source SUMO model of Ann Arbor was developed; (4) Eco-Routing algorithms in Ann Arbor using the SUMO model shows 6% fuel saving potential; (5) Experiments conducted at the Mcity test facility shows that human drivers roughly follow the Eco-driving suggestions roughly 70% of the time; (6) Based in the Ann Arbor travel patterns, we found that each shared automated vehicle can replace around 4 individually owned vehicles; and (7) Adaptive Traffic Signal Control Algorithm developed through this project has been validated both in simulations and preliminary test results. For connected and automated vehicles, on average the performance is 13% delay reduction, and 10% fuel reduction.

While connected and automated vehicles are in their early stage of deployment, the results from this project confirm that there is significant potential for energy saving if the technologies are developed and used properly. The three main technologies studied in this project include eco-routing, shared autonomous rides, and adaptive traffic signal controls. The data collected and model developed through the project can be used to study many other connected and automated vehicle technologies.

Accomplishments

This project consists of five inter-connected tasks. The achievements and update on the status of the milestones are reported for these five tasks separately below.

Task 1 Instrumentation and data acquisition of energy related information

Task 1 is the portion of the project in which data loggers are installed on 500 vehicles in the Ann Arbor area in order to generate data that will be used to calibrate the Polaris energy-consumption model (Task 4), in addition to providing naturalistic driving and energy use data that will be used in other tasks. Data collection and fusion are also developed in other tasks, but Task 1 focuses on the driving data collection and fusion of those data with other data.

Overall, Task 1 is almost complete. The remaining work is to monitor the vehicles in the field and eventually remove the data collection equipment from the vehicles. The task will be complete at the end of December 2018. Details of the six subtasks follow.

The table shows the six subtasks and delivery times for Task 1, with a note regarding the expected completion time. Four of the subtasks are complete, another (Subtask 1.5) is on time, but the Subtask 1.6, which addresses the actual fielding of the devices on 500 vehicles, is continuing. The data collection has been proceeding for well over a year.

Subtask	Subtask	Schedule	Schedule end date	Status
1.1	Coordinate and prioritize vehicle signals	M1	12/5/2015	Complete
1.2	Main data acquisition/ storage/ transmission hardware development and OBD port logger	M2-M12	11/5/2016	Complete
1.3	Decode and translate vehicle energy usage information for logging	M5-M7	6/5/2016	Complete
1.4	Integrate safety and other DAQ systems into main aggregation equipment	M5-M10	9/5/2016	Complete
1.5	Integrate driver choice/behavioral model (routing) information	(See Task 4 section)		
1.6	Data collection, instrumentation refinement, and QC	M8-M36	Complete Nov 15, 2018	Complete

Progress and achievements during the most recent quarter of the project are described below, by subtask.

Subtask 1.1. Coordinate and prioritize vehicle signals

This subtask has been reported upon in previous status reports, and is complete.

Subtask 1.2 Main data acquisition/storage/transmission hardware development and OBD port logger

Subtask 1.2 is considered complete.

Previous project status reports introduced the data logger being used, which is a FleetCarma C2 OBD-II connector device that works with combustion only, hybrid, and pure electric vehicles. We have 500 devices and have installed devices into 509 vehicles, including 16 plug-in electric vehicles. (See more on current device status in Subtask 1.6.)

Subtask 1.3 Decode and translate vehicle energy usage information for logging

Subtask 1.3 is complete.

The data collected for energy usage has been modified as the project has progressed, based on our growing understanding and negotiations with the data logger vendor. This been reported on in previous quarterly reports, and will not be repeated here.

Subtask 1.4 Integrate safety and other DAQ systems into main aggregation equipment

This project was completed before this quarter, but a summary follows for convenience.

This subtask addresses the need to fuse data from multiple sources into a comprehensive data set to support analyses for the project. The sources include:

- Onboard OBD-II logger (basic travel, location, time, and energy-related variables for 500 vehicles),
- Onboard DSRC devices (basic travel, location, time on some DOE project vehicles, and onboard a few thousand vehicles from other ongoing projects, as appropriate),
- Metadata for vehicles and traffic signals (e.g., driver demographics, roadway network)

This subtask is considered complete, as the method for fusing the data is known and similar fusion has been done at UMTRI in the past.

Subtask 1.5 Integrate driver choice/behavioral model (routing) information

This subtask involves providing an in-vehicle system with information that would provide driver information on the congestion they may face on a planned trip, and advise a route. This effort is reported in Task 4.

Subtask 1.6: Data collection, instrumentation refinement, and QC

This subtask involves the outfitting of the 500 vehicles with the ODB-II logger, validation of the system – including the backhaul – and maintaining operations. To date, 7.7 million miles of data have been collected in this study, and data collection continues. To date, over 787,000 ignition cycles have been collected across 528 vehicles.

At this time, there are still 279 units installed and actively providing data. The devices are used until the cellular data package expires, which has occurred over the past months for over one third of the devices. The remaining devices' cellular data packages expire early in November, so that data collection was completed in November 15, 2018.

UMTRI has provided a copy of the database to date to the Argonne National Laboratory (ANL). This is the complete set of data. This is possible because the informed consent form that the vehicle drivers signed stated allows this data transfer, and ANL also agreed in writing to protect the data in accordance to the privacy and data security requirements. The team is considering other ways to share data while complying with the informed consent agreements with the drivers, including sing a three-step approach to protect the starting and ending point of all trips.

Task 2 Display energy related information to study its influence on the driver

Task 2 is designed to understand drivers' behavior and decision making when interacting with vehicle connectivity technologies through a set of experiments. In the first experiment (task 2.1-task 2.4), research team developed a flexible driver information module capable of displaying a variety of information to the vehicle operator. Information relevant to the energy savings functions will be displayed to the driver and recorded so that this information can be used for later analysis. Through a display screen, we will display information (i.e., countdown to traffic signal change from green to red) to allow for a range of vehicle as well as driver efficiency coaching messages. The type and amount of information displayed to the driver will be adjusted to aid in the assessment of various driver information strategies. In task 2.5, factors impacting on drivers' decision making in choosing routes are examined through a field study.

The table shows the proposed subtasks and delivery times for the five subtasks of Task 2. The starting date of the project is taken to be November 5, 2015, the date of the kickoff meeting.

Subtask	Subtask	Schedule	Schedule end date	Status
2.1	Identify CAV user functions, co-design and prioritize signals	M1	12/5/2015	Complete
2.2	Develop driver information display hardware and communication	M2-M7	6/6/2016	Complete
2.3	Design vehicle information display screen(s) and experimental cases	M8-M12	12/7/2016	Completed
2.4	Review of the finished human test results. Review the field performance of the designed user interface	M13-M24	12/30/2018	Completed
2.5	Driver route choice survey and guided-directions application	M30	11/30/2019	Completed

Subtask 2.1. Identify CAV user functions, co-design and prioritize signals

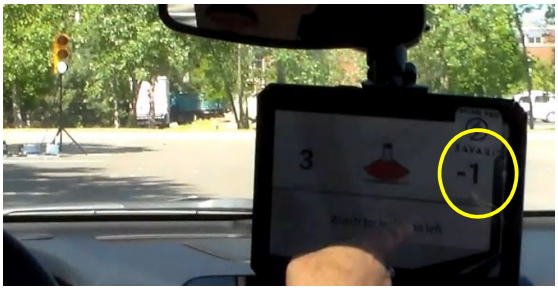
Early on in the project two key CAV functions were identified that both show significant potential, and had not been implemented in large-scale experiments: eco approach and departure, and green wave (speed recommendation).

This subtask is considered complete.

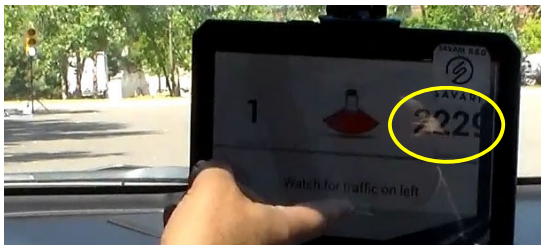
Subtask 2.2. Develop driver information display hardware and communication

UMTRI has worked with Savari Inc. for developing hardware, software, and communication solutions for the in-vehicle SPaT (Signal Phase and Timing) visualization and speed recommendation at intersections. The hardware include ASD (Aftermarket Safety Device), antennas, DVI display (tablet), WIFI dongle, and USB drive. Figure 2.1 shows the driver information display of the prototype system that was demonstrated at the end of June 2016. Note this prototype is mainly focused on the system functionalities rather than the HMI. The demo HMI is rather crude and has been improved during Subtask 2.3 for the final testing.

(a) Recommended speed



(b) No recommended speed needed



(c) Cannot pass on green, prepare to stop

(d) Red light violation

Figure 2.1 Savari prototype recommended speed application

Subtask 2.2 is considered complete.

Subtask 2.3 Design vehicle information display screen(s), vehicle instrumentation and data collection

During this period of report, tasks on data collection, data reduction and data analysis have been completed. The objective of this testing is to identify drivers' reaction and acceptance to the energy related feedback system. An experiment has been conducted tin controlled field environment (MCity testing facility). The intersections at Mcity are equipped with DSRC Road Side Units (RSU) by using SAE J2735 standard to allow communication between the vehicles and the infrastructure. For Wireless Access in Vehicular Environments (WAVE), the SAE J2735 Standard specified a message set to utilize the 5.9 GHz Dedicated Short Range Communications (DSRC) (Iglesias et al., 2013). Drivers followed the course outlined in blue arrows (Figure 2.2). Data collection centered around the run from the green cone to the red stop line. In-vehicle tablet device began receiving SPAT information from the RSU around orange box while the tablet began receiving SPAT data from RSU at about 100 meters out from stop line.

The in-vehicle DSRC-based V2I devices, including both hardware and software pieces, were developed through a collaboration between the University of Michigan Transportation Research Institute and Savari Inc. The V2I devices received and displayed real-time SPAT information and provided speed recommendations based on both the current SPAT information and the vehicle's travelling speed. The devices included an Aftermarket Safety Device (ASD), antennas for ASD, a tablet screen, a WI-FI dongle, a GPS receiver, and a USB drive (See Figure 2.3). Three cameras were used to record the vehicles' front view, drivers' face, and over the shoulder view. When the vehicle was approaching an intersection, information relevant to the energy-saving and safety-improving functions (e.g., SPAT information and recommended speed) was displayed to drivers on the tablet screen. The human–machine interface (HMI) tablet provided drivers with real-time information such as the current speed, suggested speed, and the countdown to traffic signal change from green to red (See Figure 2.4).

Figure 2.2 MCity test procedure

Fig. 2.3 Vehicle instrumentation and HMI display

In the experiment, a total of 32 participants from two age groups, 16 younger (20-30 years old) and 16 middle-aged (40-50 years old), were recruited through a random selection method. Gender was balanced within each age group. To avoid the influence of novice drivers' lower confidence and skill levels on their driving behavior, all the participants had at least five years' driving experience (mean=18 years, SD=10, range 6-33). Their education levels were divided into three groups: some college or lower (19%), bachelor's degree (47%), and master's degree or higher (34%).

As shown in Figure 2.2, each driver was instructed to follow the course outlined in blue arrows during the experiment a total of 15 times: a practice drive, seven baseline drives without the DSRC device activated, and seven treatment drives with the DSRC devices activated. Data collection was geo-fenced by using the area from the green cone to the red stop line, to capture intersection approaching segments. Drivers were told to arrive at the green cone with a speed of close to 25 mph (40km/h). DSCR devices began receiving SPAT information from the RSU around the orange box, which was located about 100 meters out from the red stop line (i.e., entrance of intersection). After getting familiar with the testing field, each participant experienced seven different intersection scenarios in both baseline and treatment conditions:

Scenario 1: "Green Same Speed" — Current phase is green, and the vehicle can pass on green with current speed — No speed changes required to pass the green phase;

Scenario 2: "Green Speed Up" — Current phase is not green, and it requires acceleration to pass green light phases — Required speeding up to pass the green phase;

Scenario 3: "Green Slow Down" — Current phase is not green, and the vehicle can pass on green if change its current speed by decelerating — Required slowing down to pass the upcoming green phase;
Scenario 4: "Green Stop" — Current phase is green, and the vehicle cannot pass on green under any conditions — Impossible to pass green phase;
Scenario 5: "Red Through" — Current phase is red, and the vehicle can pass on the next green if it changes its current speed — No speed changes required to pass the upcoming green phase;
Scenario 6: "Yellow Stop" —Yellow dilemma zone (less than 8 seconds left for the green light phase) — Impossible to go through (i.e., make the yellow light);
Scenario 7: "Yellow Through" — Yellow dilemma zone (more than 10 seconds left for the green light phase) — Possible to go through.

Both objective data (e.g., driving speed, throttle position, brake pedal use) and subjective data (e.g., user acceptance from questionnaires) were collected and analyzed in this study. A post-study questionnaire with 14 questions was distributed to all participants to collect their opinions and acceptance of the system.

Fig. 2.4 Example of video views and information displayed through HMI

Subtask 2.3 is considered complete.

Subtask 2.4 Data analysis and interpretation

The research team has completed the main part of this subtask and one manuscript is accepted and will be presented at the 2019 Transportation Research Board Annual Meeting. The purpose of this analysis is to examine and predict whether and how drivers will change their behavior when provided with energy- and safety-related strategy recommendations from V2I communication. Contributing factors to the potential behavior changes are explored from four aspects, including vehicle kinematic features, device information, driver characteristics, and subjective data.

Results on drivers' acceptance

To analyze how much drivers will change their behavior when they are offered speed suggestions, the compliance rate of recommended speed choices in the treatment group was compared with the result in the baseline, as shown in the orange bars in Figure 2.5. In the baseline, there were no suggested speed strategies, so recommended speed choices in the treatment group were used as comparison references, after matching the constraints that include the same driver, same scenario, similar driving speed (speed variation within 5 mph) when DSRC devices began to receive SPAT information. Finally, 201 valid samples from the baseline group were obtained. Results showed that in general, the compliance rate during the treatment drive was much higher than in the baseline. Under scenario 2 ("Green Speed Up"), drivers' behavior had the largest change with a 42% increase in the compliance

rate after drivers received suggestions from the V2I device. The second largest increase was 36% under scenario 5 ("Red Through"), where drivers were more likely to keep current speed to enter the intersection when they were provided with suggestions. In addition, compared with the baseline, compliance rates under both scenario 1 ("Green Same Speed") and 7 ("Yellow Through") increased more than 20% recommended speed choices. However, a decline of the compliance rate occurred under scenario 4 ("Green Stop"). This might be because when information of the countdown to traffic signal change was provided, drivers were aware that they had a chance to enter the intersection in the next green phase if they decrease the driving speed dramatically and then approach the intersection at an extremely low speed.

Mixed model analyses were performed by using the PROC GLIMMIX procedure in the statistical software package SAS 9.2. Drivers' compliance rate was significantly affected by scenarios ($F(6, 380) = 8.70$, $p<0.01$), offering suggestions or not ($F(1, 380) = 12.95$, $p<0.01$), and age ($F(1, 30) = 4.18$, $p<0.05$). Younger drivers showed an 11% higher average compliance rate than middle-aged drivers. In addition, the interaction between scenario and offering suggestions or not ($F(6, 380) = 3.73$, $p<0.05$) also led to obvious changes of the compliance rate.

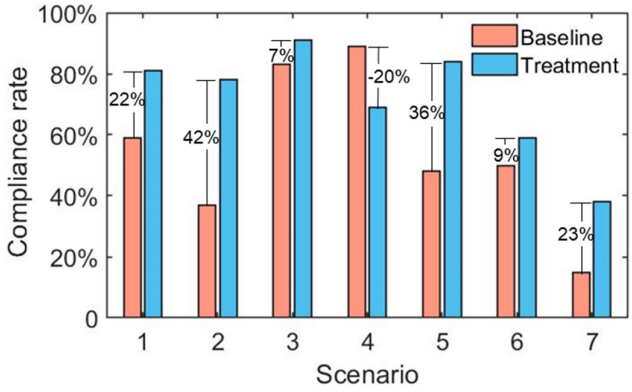


Fig. 2.5 Drivers' compliance rate of recommended speed choices under seven scenarios

Results on predicting drivers' reactions when offered speed strategies

To predict drivers' reactions towards speed choice recommendations, The Random Forests Algorithm was applied by using the 224 samples in the treatment group were used. There were 13 independent variables, from four sources:

- **Vehicle kinematic features**, obtained at locations where DSRC devices begin to receive SPAT information from the RSU: current driving speed, throttle position, and brake;
- **Device information**: scenario, and V2I suggestions;
- **Driver characteristics**: age, gender, driving experience, and education level;
- **Subjective data:** Q1 (reliability and usefulness), Q2 (user-friendliness), Q3 (suggestion receiving and satisfaction), and Q4 (driving risk).

The detailed definitions of the input variables, together with their distributions, are presented in Table 2.1. The dependent variable is drivers' reactions, namely, drivers' behavior approaching the intersection. The distribution of the dependent variable is as follows: no change (30%), acceleration (18%), deceleration (without completely stopping) (26%), and stopping (26%).

All 224 samples were split randomly into two sets by the ratio of 7:3 for training and testing respectively. A Random Forests model was built by inputting 13 independent variables and run in the R software. According to the stable and minimum values of the OOB error rate, ntree and mtry were set to 600 and 6 respectively. The prediction result of Random Forests was shown in Table 2.2. In the

training group, the OOB error rate was 27.7%, namely, the OOB accuracy was 72.3%. In addition, the overall testing accuracy was 75.8% (95% CI: 63.3-85.8%). This Random Forests model performed well in predicting the four-category variable with a relatively high accuracy. Using this model, drivers' reactions at the intersection could be predicted by the data obtained about 87.4m (SD=2.4, range 75.5-92.8) away where the vehicle started to receive signal phasing and timing information.

Table 2.1. Definitions and distributions of input variables

Input Variables	Description (Units)	Min	Max	Mean	S.D.
Vehicle variables (obtain at the position where DSRC devices begin receiving SPAT information)					
Speed	Driving speed (mph)	20	35	26	2.6
Throttle position	Position of the accelerator pedal collected from the vehicle network and normalized using manufacture specs (%)	0	17	1.9	3.1
Brake	Pressing the brake pedal=1; otherwise 0	1 (6%); 0(94%)			
Device information					
Scenario	Seven different intersection scenarios	Each scenario (14%)			
V2I Suggestions	Recommended speed choices	No change (30%); Acceleration (12%); Deceleration (15%); Stopping (43%)			
Driver characteristics					
Age	Younger:20-30; Middle-aged: 40-50	Younger (50%); Middle-aged (50%)			
Gender	Gender of drivers	Male (50%); Female (50%)			
Education	Education levels of drivers	Some college or lower (19%); Bachelor (47%); Master or higher (34%)			
Driving experience	Years of driving experience (years)	6	33	18	10
Subjective data					
Q1	Reliability and usefulness	8	16	13	2.1
Q2	User-friendliness	8	17	14	2.3
Q3	Suggestion receiving and satisfaction	-1	6	3	1.7
Q4	Driving risk	1	6	4	1.1

Table 2.2. Prediction results of Random Forests

Training Set						
Predicted	Observed				OOB error rate	OOB accuracy
	No change	Acceleration	Deceleration	Stop		
No change	39	3	4	1	27.7%	72.3%
Acceleration	4	21	3	2		
Deceleration	5	1	29	10		
Stopping	0	4	8	28		

Testing Set					
Predicted	Observed				Accuracy
	No change	Acceleration	Deceleration	Stop	
No change	18	2	0	0	75.8% (95% CI: 63.3%~85.8%)
Acceleration	1	7	0	0	
Deceleration	0	1	12	7	
Stopping	2	0	2	10	

This subtasks is 100% complete.

Subtask 2.5 Driver route choice survey and guided-directions application

The purpose of task 2.5 is to collect naturalistic driving data and understand factors associated with drivers' decision making on route choices. This task 2.5 contains four main subtasks which are all completed: (1) EcoRouting app development, (2) experiment design and conduction, (3) data analysis and results interpretation, and (4) final report preparation. ANL has developed the EcoRouting app that was used in this study and collaborated on preparing with the final report. UMTRI research team has led the effort of subtask (2), (3), and (4). The research team has recruited a total of 43 participants in this study and the final dataset was identified from 39 participants and used in the analysis, as some GPS data recording or missing data issues occurred during the data collection of four participants. The methods and results are summarized in the following sections.

1. Introduction

The main goal of this turn-by-turn navigation app was to collect all the answers to the questions of a survey presented at the beginning of the route, the number of detours with all the new created directions and all the GPS coordinates followed by the driver.

The app was able to provide three different routes to the driver varying the time and the fuel consumption on the routes (eco vs fast vs "balanced"). The app was also able to provide the turn-by-turn guidance using the selected route while recording the data gathered along the route. The route and navigation services were provided by the **MapQuest API**. This API offered the route calculation after the input of the desired destination and also provided all the data displayed in the turn-by-turn navigation screen.

The recording was made locally in a database file using **SQLite API**. In the code of the app, there were two different databases, one was called '*Trips.db'* and the other was '*cars.db'*. The database were all the data for the analysis was recorded is '*Trips.db'*. The '*cars.db'* file just recorded the different cars used in order to introduce corrections in the required fuel estimation made by the app at the beginning of the route.

2. Fuel consumption estimation

The three provided routes displayed an estimation of the fuel consumption that was calculated considering two main effects: the **average velocity**, the **traffic** and the **car used**. The **effect of the velocity** was modeled making use of the data provided by the user, since he was supposed to provide the MPG value in the UDDS cycle (average velocity of 21.2 mph) and in the HWY cycle (average velocity of 48.3 mph), these two values are easily found in the EPA website for **different car models**. In https://www.fueleconomy.gov/feg/driveHabits.jsp, it is explained that the maximum MPG occurs around 40-50 mph with a low effect of the speed in this area. Also, when the speed is 80 mph there is a reduction of around 35% in the maximum value of the MPG. Knowing this and using the two provided values of MPG, the dependency of MPG with the speed observed in Figure 2.5.1 is obtained.

The **effect of the traffic** was modeled using the data provided by MapQuest, since they were able to classify different segments of the route according to the traffic. The three main classes of traffic segments were FREE_FLOW, SLOW, and STOP_AND_GO.

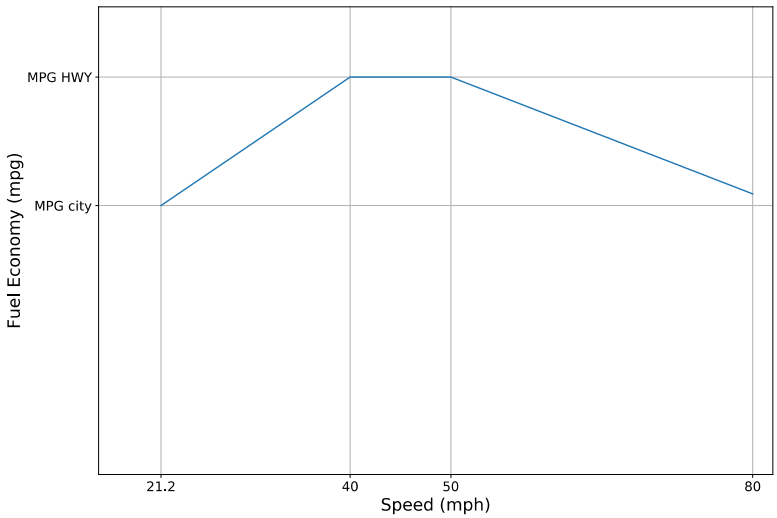


Figure 2.5. 1 Effect of the speed on the MPG during a highway driving

With these two main effects on the MPG, the calculation was made using the following procedure after splitting the whole route into different segments according to:

1. The whole route was split into different segments according to the traffic classification made by MapQuest.
2. Calculation of the average MPG on each route segment with FREE_FLOW traffic was made knowing the average speed in the segment and using the plot of Figure 2.5.1.
3. The global MPG of the route was estimated assuming that the vehicle had the value calculated before in the FREE_FLOW segments of the route, and the city cycle value in the STOP_AND_GO and SLOW segments. The value of all these segments were averaged using their distances.

$$MPG_{global} = \frac{\sum Distances_{FREE\_FLOW} * MPG(Average\ speed) + \sum Distances_{NON\_FREE\_FLOW} * MPG\_city}{Total\ distance}$$

3. App Functionality

The first screen (Figure 2.5.2) that appears in the app is a list with all the recorded trips, from this screen the user can access to the information gathered on each trip, and create a new one. When the *'plus'* button is pressed, a screen to select the car that will be used appears.

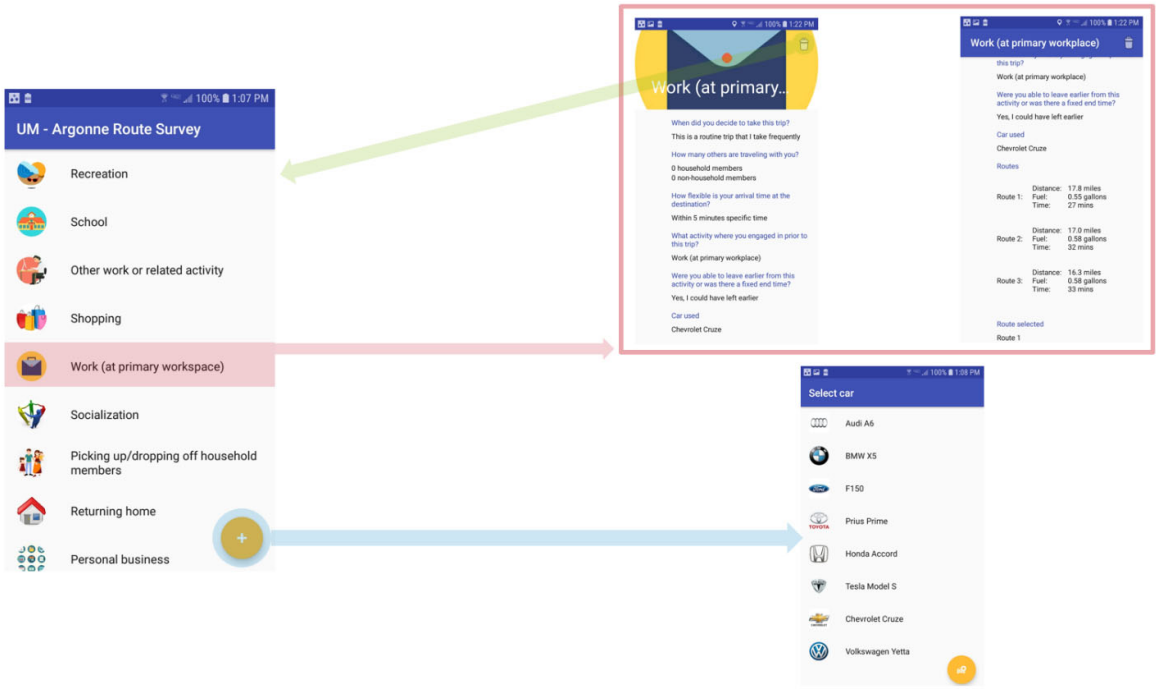


Figure 2.5.2. List of trips, trip details and select car

Once in the screen to select the car, the user can create a new car with its MPG values or directly select a previously used car to use it during the route or to edit it (Figure 2.5.3). When the car has been selected, the user accesses to the survey screen where all the questions to answer are displayed (Figure 2.5.4).

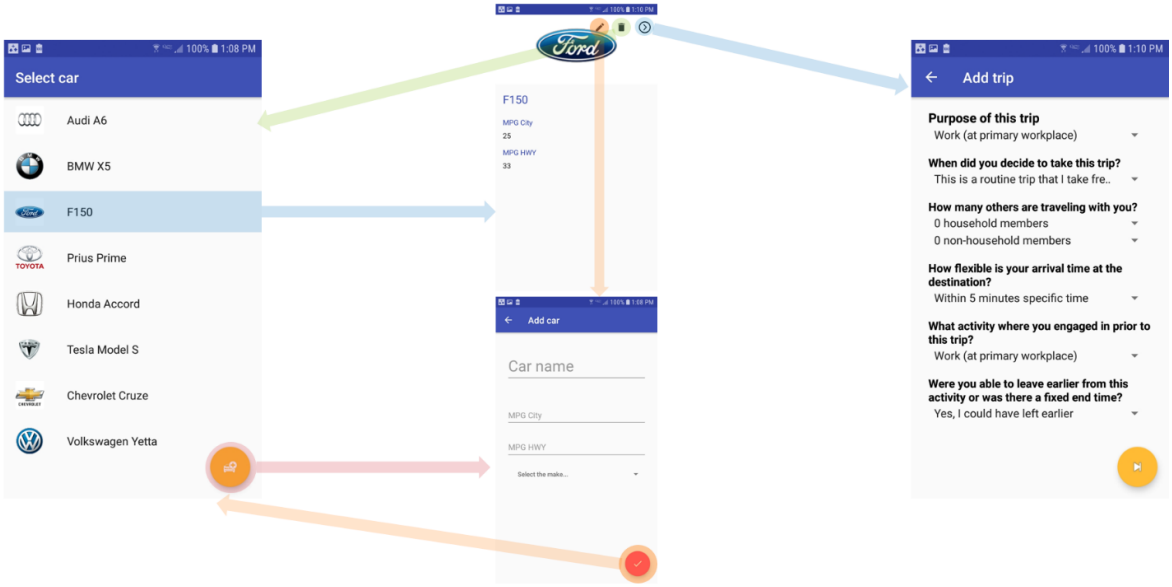


Figure 2.5.3. List of cars, create new car, edit car, and survey

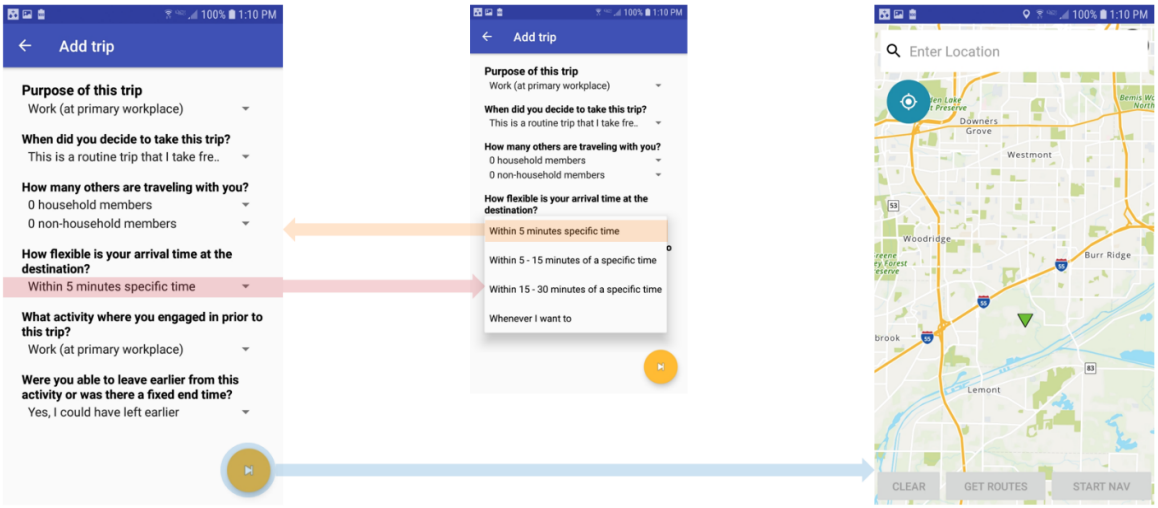


Figure 2.5.4. Answering the questions

When the user has answered all the questions, the route selection screen appears. On this screen, the user can type the desired destination and three routes appear with different values for the distance, required time to complete and required fuel. The user can select the desired route and pressing the *'START NAVIGATION'* button, the navigation screen will start (Figure 2.5.5).

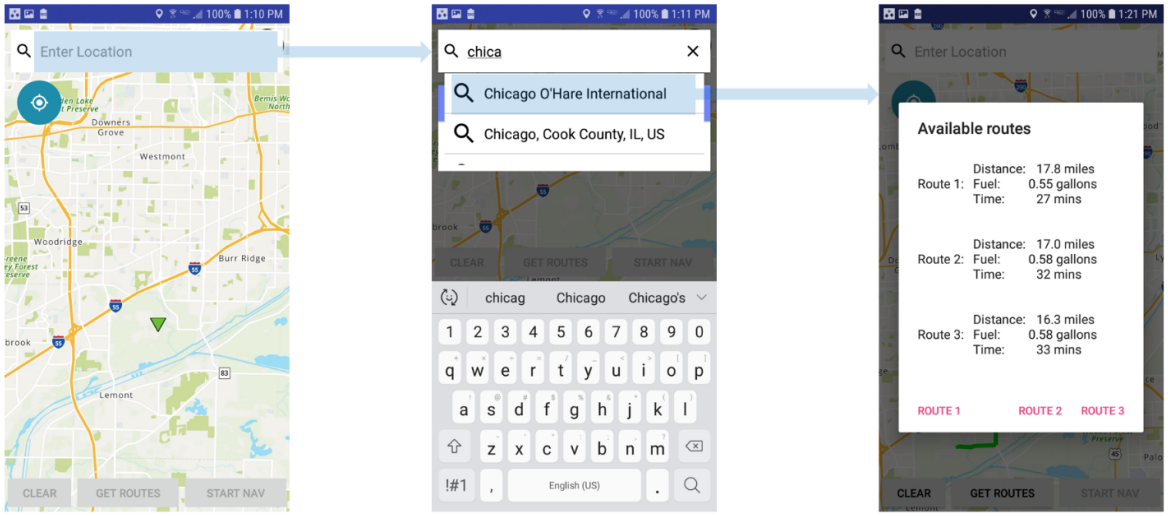


Figure 2.5.5. Route selection

The Navigation display shows the meters left to complete the route with an estimated time to arrive. The user can also find all the directions that he will find by pressing the *'List'* button (Figure 2.5.6). In addition, it has to be noticed that each time that the *'Pause'* button was pressed, an instance of the trip was generated.

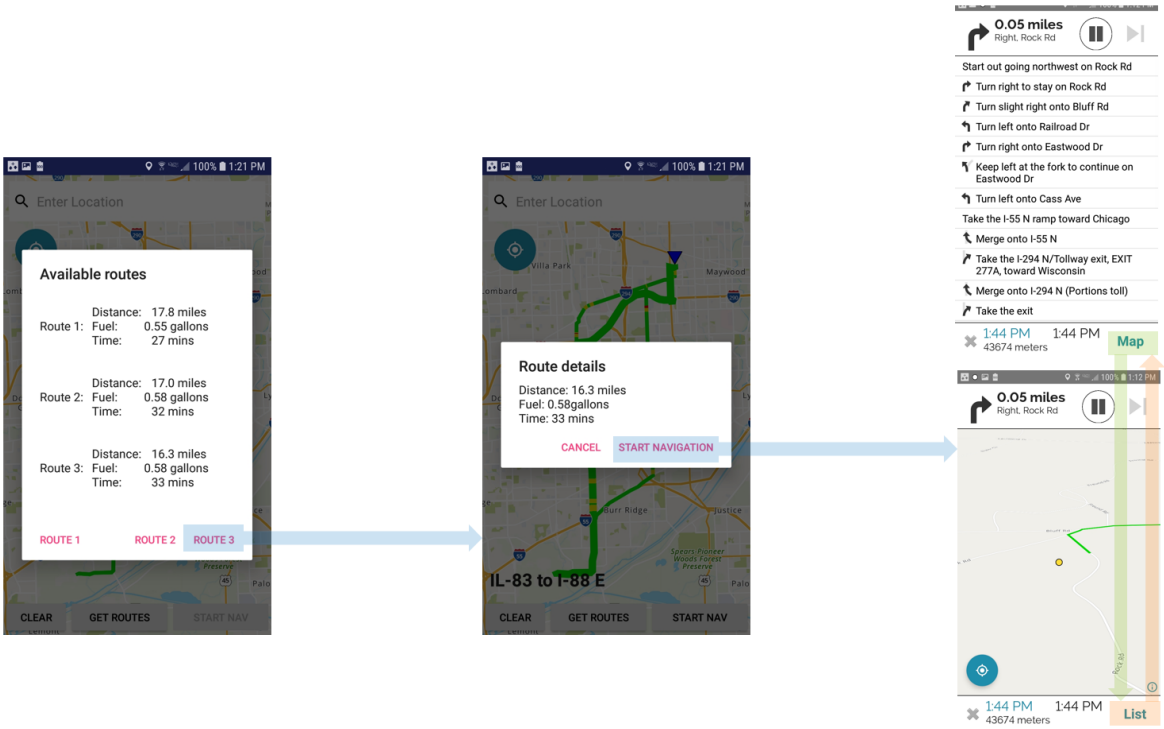


Figure 2.5.6. Navigation screen

Data Collection

Data Collection

Forty-three participants were given a cell phone handset for two weeks and asked to interact with a custom-designed and built "Eco-Routing" application. This interaction included completing a route-choice survey and selecting from recommended driving routes prior to beginning of some of their driving trips over a 2-week period. Participants who were also participating in UMTRI's Ann Arbor Connected Vehicle Test Environment (AACVTE) were recruited to receive the cellular device with the custom-designed, Eco-routing software application (Eco-routing device.)

Participants first came to UMTRI to pick-up the Eco-routing Device and to learn how to operate it. Upon arriving at UMTRI for their visit, participants were presented with the requirements for participating in the Eco-routing effort. They reviewed and completed the Eco-routing Informed Consent document if they accepted the terms of the project. The expectations for the interaction process with the Eco-routing cell phone application were then explained to the participant. They were walked through setting up the application, inputting a trip, choosing a suggested route and using the navigation function.

Application

Set-up for the Eco-routing application on the cell phone included having the participant name their vehicle (in case they used more than one vehicle,) and input the gas mileage they experience in their personal vehicle for both city and highway driving. Then they were walked through the pre-drive survey questions and asked if they needed any clarification. Finally they

were shown how to initiate a new trip, and enter a destination into the application. They could either tap on a map on the screen or directly input an address to enter the location of the desired destination.

Before each trip began, the interaction with the cell phone application included answering questions about the nature of the trip and the participant's trip planning process via a short, eight question survey embedded in the application. After answering a few of these survey questions, participants then input their desired destination for their planned trip. Next, the application offered them one, two or three different driving routes to their destination from which to choose. Each suggested route was listed with its expected distance, expected fuel consumption and expected time duration. Participants were instructed: "The application will provide you with the distance, time and estimated fuel consumption for up to three routes. Choose the route that is most desirable to you (Figure 2.5.7). It is likely that one of these routes is the best route for you to take, but you are not required to follow them." After choosing a route, the device provided route guidance to the participant via both live turn-by-turn directions on the screen and via verbal directions from the cell phone speaker. Maps and navigation used the MapQuest API.

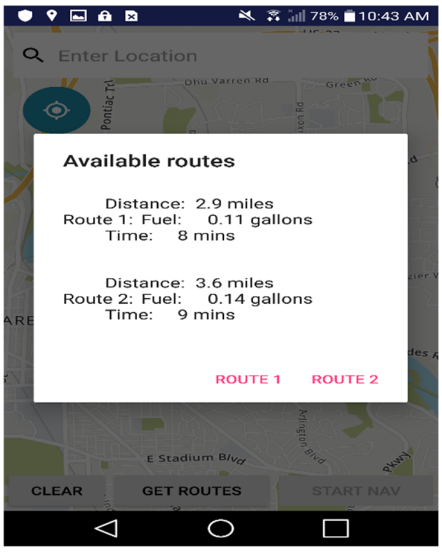


Figure 2.5.7. Screenshot of Route Selection Screen

The number of suggested routes was a function of the complexity of the route. For a very short route, there was likely to be only one reasonable route as this would be the quickest temporally, the shortest distance-wise and most economical for fuel consumption. For more complex routes there could be more fundamental differences between route choices, often such as whether to take the highway or surface streets, where to get on or off of the highway, or sometimes two opposing directions around a city area.

Figure 2.5. 8. Treatment Route (actual GPS data from the participant,) and Recommended Route (trace route created in Ovitalmaps.) This is an example of a simple route for which the participant did not follow the Recommended Route.

Once the participant arrived at the destination they received a message indicating as such. If they deviated from the Recommended Route directions while driving, new directions were created instantly, in real-time starting from the current location (treated as a new starting location) and presented to the participant through the interface (as well as recorded in the data file.), as shown in Figure 2.5.8.

The application displayed how many trips the participant had input data for and "completed." Participants were expected to record data for 20 trips over their two-week exposure period. At the end of the exposure period participants returned the device and UMTRI researchers downloaded the data to UMTRI servers. Participants were paid $100 for their two-week participation if they completed the survey for all 20 trips and did at least enough driving that the application gave them credit for completing a trip.

Data Collected

A total of 43 participants were originally recruited. The data included the responses to the questionnaire questions for each trip as well as the route chosen and the GPS data collected on the device for the specific trips.

Data Reduction

Data was parsed and loaded onto UMTRI servers and was generally accessed using Microsoft Transact-SQL. Data was aggregated and analyzed to determine which trips were "Valid" in the sense that they were comprised of complete and correctly collected data from a participant.

Valid trips were first identified as having a completed survey and a fairly complete GPS dataset. Trips that were duplicates of other trips (potentially where a participant entered the same trip twice) were also flagged as "invalid." Finally, later in the analysis if it was found that the participant did not travel to their inputted destination, the trip was also flagged as invalid and removed from the dataset used for this analysis.

The GPS data from the Treatment Trips was mapped for each valid trip for each participant to produce the Treatment Route Maps. Data for the Recommended Route was in the form of turn-by-turn directions, so in order to compare the two routes, a trace map was made for each valid Recommended Route chosen by a participant before a Treatment Trip. The Ovitalmaps software was used to create the trace maps of the Recommended Route. These trace maps were created by stepping through the turn-by-turn directions and placing pins at critical points on a map to create a trace of the Recommended Route. Researchers creating the trace maps began with the GPS location of the beginning and end of each trip and connected the two points.

To determine if the participant followed the Recommended Route for a given trip, the Treatment Route and the Recommended Route trace map were visually inspected to determine if the exact same roads and turns that were recommended in the Recommended Route were actually followed in the Treatment Route. Any deviation on surface streets or highways from the Recommended Route would result in a trip being scored as "Did not Follow Route." Slight deviations within parking lots, shopping centers, apartment communities and small, unmarked subdivisions at the beginning and end of a trip were outside of the scope of directions and would not result in a route being scored "Did not Follow Route." Table 2.5.1 presents a summary of the data collected. Forty-three participants received an Eco-Routing device. Valid Trips were collected from thirty-nine participants. Participants on average recorded data for 26.8 trips but only contributed valid data for 18.9 trips. Overall, participants followed the Recommended Route in 434 of 738 valid trips or 58.8 percent of valid trips.

Table 2.5.1. Information on Participants' Valid trips and route choices

Participant ID	All Surveys	Valid Trips	Followed Route	Did Not Follow
1	25	12	7	5
2	27	17	7	10
3	21	17	6	11
6	21	14	11	3
7	22	22	14	8
8	22	19	13	6
10	16	15	8	7
11	31	14	8	6
12	51	30	16	14
13	29	15	4	11
14	28	24	11	13
15	20	16	9	7
16	26	19	14	5
17	40	37	26	11
18	28	20	15	5
19	24	19	12	7
20	25	22	11	11
21	6	3	0	3
22	69	20	17	3

23	29	10	7	3
24	37	23	15	8
25	25	22	15	7
26	50	46	29	17
27	32	13	4	9
28	21	18	11	7
29	27	18	15	3
30	20	19	9	10
31	20	15	9	6
32	21	20	18	2
33	20	17	8	9
34	22	22	13	9
35	39	24	10	14
36	22	20	12	8
37	24	20	8	12
38	20	12	10	2
39	20	17	10	7
41	19	11	2	9
42	32	24	13	11
43	16	12	7	5
Total	1047	738	434	304
Mean	26.8	18.9	11.1	7.8

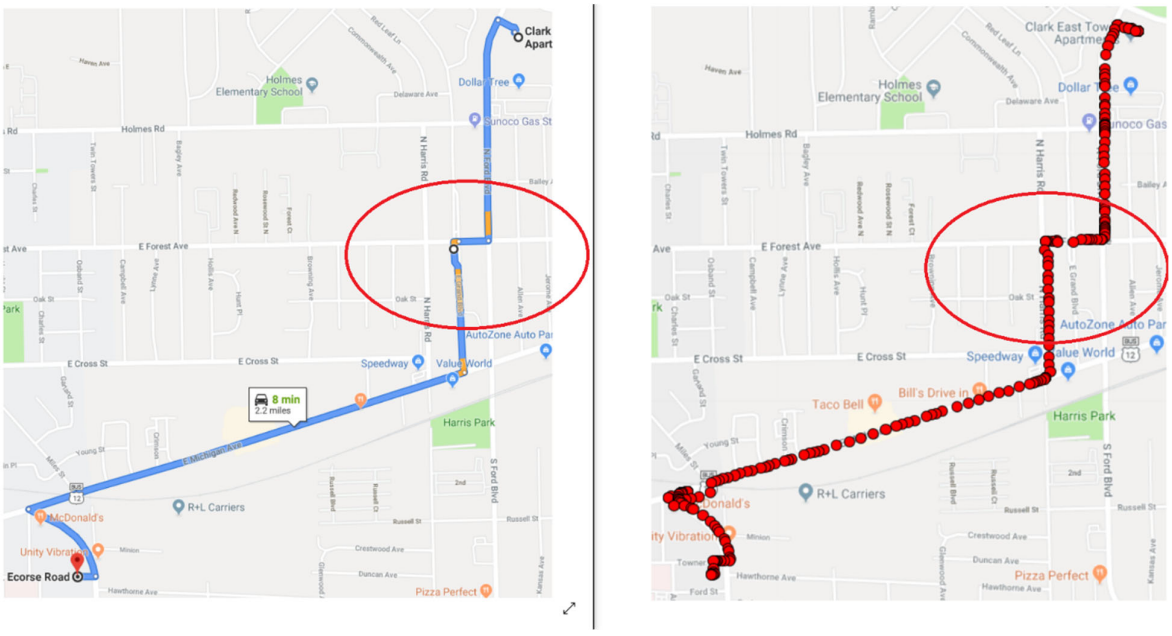


Figure 2.5. 9 One example of "followed trip"

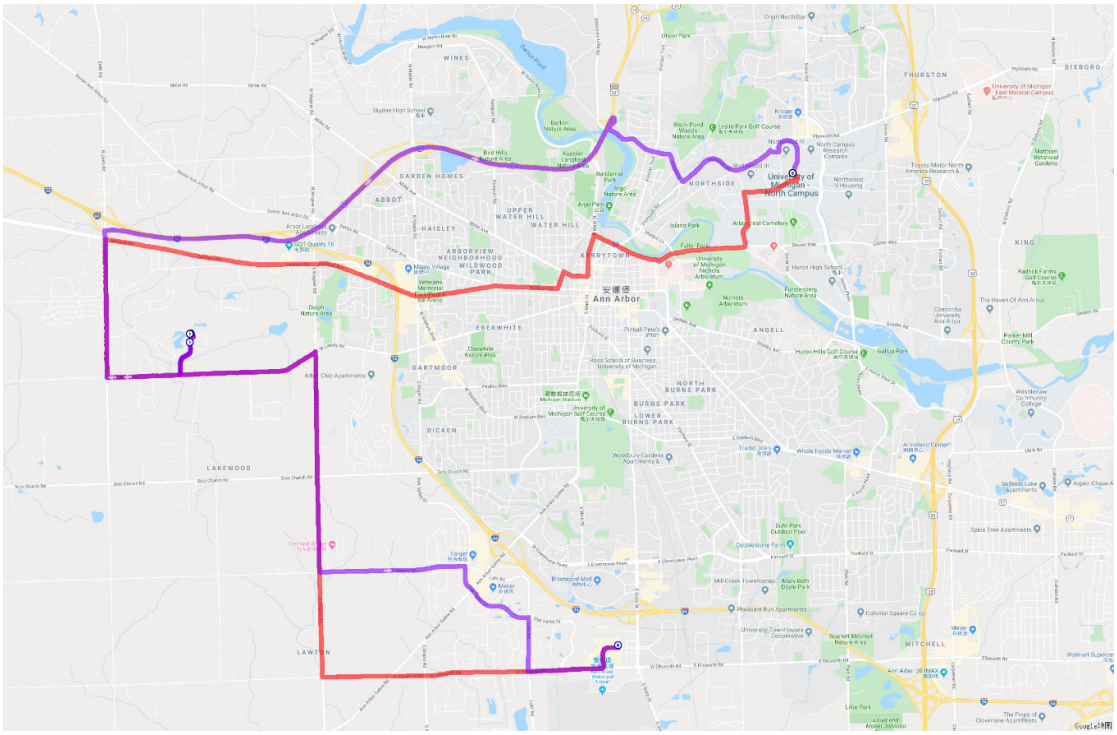


Figure 2.5.10 An example of "un-followed trip" (red is recommended route, purple is actual driving)

Data Analysis and Results

There are two main purposes in this analysis: One is to investigate and predict what kind of recommended route drivers will choose from eco, fast, and balanced options; The other is to explore whether and why drivers will follow the recommended route. Impacting factors on the driver's route choices and following are examined from three aspects, including driver characteristics, subjective data, and route information.

Modelling driver's route choice

Since each recommended route may have several different features at the same time, for example one recommended route is the most fuel efficient as well as the fast one, the driver's route choice is a multi-label problem. To mitigate the effects of unbalanced sample sizes for different drivers, the average probability of route choice was calculated based on each driver, as shown in Figure 2.5.11. In general, drivers were more likely to choose the fast route, having the highest average probability 83.4%, followed by the eco route with a selection probability around 78.6%, while the routes with the balanced feature had the least likelihood to be selected, averagely 70.7% for each driver. There were totally 18 variables coded for each trip, derived from driver characteristics, subjective data, and route information. After eliminating the highly correlated variables, such as gas consumption and driving time of the selected route were excluded due to their high correlations with distance, 14 variables were finally chosen as the input variables for further analysis. Table 2.5.2 demonstrates the detailed descriptions and distributions for all these candidate variables.

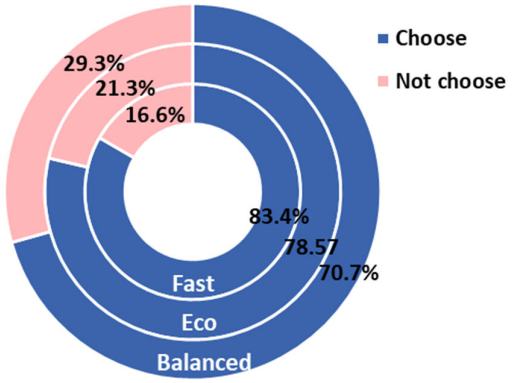


Figure 2.5. 11. The average probability of route choice

Table 2.5. 2 Definitions and distributions of input variables

Variables	Description (units)	Min	Max	Mean	S.D.
Route information					
Distance	Distance for the selected route (mile)	0.30	132.60	9.34	11.87
Distance saving	Distance differences between the longest and the shortest recommended routes (mile)	0	13.70	0.82	1.39
Average gas consumption	Gas consumption per mile for the selected route (gallon per mile)	0.02	0.09	0.04	0.01
Sequence	The recommendation sequence for different routes	1st (75.88%), 2nd (19.78%), 3rd (4.34%)			
Number of routes	The number of recommended routes	1 (36.45%), 2(47.94%), 3(15.58%)			
Driver characteristics					
Age	Younger: 20~50, Older:50~75	Younger (48.72%), Older (51.28%)			
Gender	Gender of drivers	Male (43.59%), Female (56.41%)			
Subjective data					
Purpose	Purpose of this trip	Household errands (5.01%), Personal business (9.08%), Picking up/dropping off (3.52%), Recreation (9.62%), Returning home (29.27%), Shopping (11.11%), Socialization (6.10%), School /work (19.24%), Other (7.04%)			
Decision time	When did the driver decide to take this trip?	Earlier today (10.43%), Several days or longer (8.54%), Just now (18.56%), Not sure (0.41%), Routine (56.37%), Yesterday (5.59%)			
Household passenger	How many household passengers were traveling with the driver?	0 (85.09%), 1 (9.49%), 2 (3.52%), 3 or more (1.90%)			

Non-household passenger	How many non-household passengers were traveling with the driver?	0 (95.93%), 1 (3.79%), 2 (0), 3 or more (0.27%)
Flexibility	How flexible was the driver's arrival time at the destination?	Whenever (18.70%), Within 15 - 30 mins (4.20%), Within 5 - 15 mins (9.08%), Within 5 mins (68.02%)
Prior activity	What activity where the driver engaged in prior to this trip?	Household errands (9.89%), Personal business (13.41%), Picking up/dropping off (2.30%), Recreation (9.21%), Returning home (6.37%), Shopping (10.43%), Socialization (6.64%), School/work (26.70%), Other (15.04%)
Leave earlier	Were the driver able to leave earlier from the prior activity?	Maybe (9.89%), No (23.31%), Yes (66.80%)

Eco-route choice

To further explore the impacting factors on the eco-route choice, mixed model analyses were conducted in the statistical software package SAS 9.2 by using the PROC GLIMMIX procedure. All the input variables and their interactions were chosen as the fixed effects, while individual driver and interactions between driver and any fixed effects were treated as random effects. The dependent variable was whether choose the eco route or not. After excluding the insignificant factors, the final model was shown in the Table 2.5.3 Distance had a negative impact on the eco route choice ($t(643)=-5.56$, $p<0.001$), while average gas consumption positively affected the eco route choice ($t(643)=2.35$, $p=0.019$), indicating that drivers were more likely to select the eco route when its distance was shorter and gas consumption per mile was higher. In addition, the route recommendation sequence also had a significant effect on choosing the eco route (all $p<0.001$), and giving priority to recommend the eco route could guide drivers to choose the eco way.

Table 2.5.3 Mixed model results for the eco-route choice

Effect	Estimate	Standard error	DF	t Value	Pr>|t|
Intercept	1.139	0.715	38	1.59	0.119
Distance	-0.056	0.010	643	-5.56	0.019
Average gas consumption	40.868	17.395	643	2.35	<0.001
Sequence					
1st*	0				
2nd	-1.925	0.316	52	-6.10	<0.001
3rd	-2.518	0.491	52	-5.13	<0.001

*Note: * denotes reference group for categorical variables; only significant factors were demonstrated in this Table*

To predict driver's route choice behavior, a multi-label Random Forests (RF) classification model was established by using the "scikit-learn" package in Python software (version 3.6). Those 14 variables mention above were selected as the independent variables, while the independent variable is what kind of route drivers would choose, which is a multi-label variable with three candidate features, i.e., eco, fast, and balanced. All 737 samples were

partitioned randomly into 70% for training and 30% for testing. After 5- fold cross-validation, the parameters in the multi-label Random Forests were determined, i.e., the number of trees was 550 and the number of variables considered in each split was 4. The final prediction result was shown in Table 2.5.4. In the training group, the out-of-bag (OOB) accuracy was 87.0%, and the overall testing accuracy was 79.3%. As for the prediction results in each label, their precisions were greater than 80%. For comparison, several other machine learning methods that are commonly-used multi-label classification were also tried in this study, including K-Neighbors Classifier (KNC), Support Vector Classification (SVC); Neural Network Multi-Layer Perceptron Classifier (NNMLPC). The area under the Receiver Operating Characteristic (ROC) Curve (AUC) was used to evaluate the performance of different algorithms, and the ROC curve in the multi-label classification was measured by the average value of all labels. As shown in Figure 2.5.12, the AUC of the multi-label Random Forests classification was 0.86 which was greater than others, indicating that the multi-label Random Forests classification had a better performance.

Table 2.5.4 Prediction results of the multi-label Random Forests.

Label	Precision	Recall	f1-score
Eco	0.86	0.85	0.86
Fast	0.90	0.95	0.92
Balanced	0.81	0.77	0.78
Overall accuracy	Training (OOB): 0.870; Testing: 0.793		

Note: Precision=TP/(TR+FP); Recall=TP/(TP+FN); f1-score =2(Precision*Recall) / (Precision+Recall)*

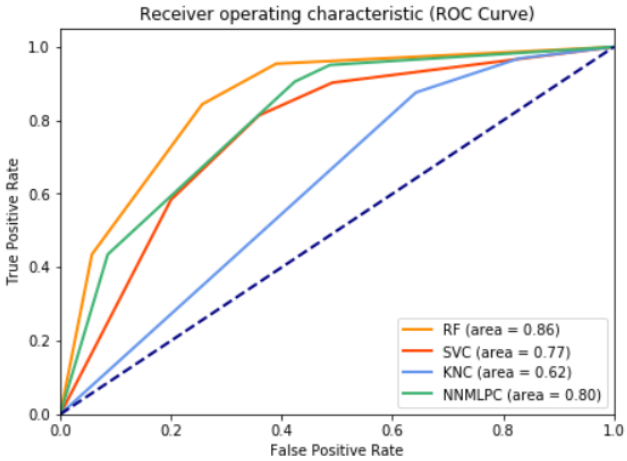


Figure 2.5.12 ROC curves for multi-label classifiers

Figure 2.5.13 illustrates the variable importance which represented the statistical prioritization of variables regarding their contribution to the prediction model. Variables from route information showed the largest impacts on the driver's route choice, i.e., distance saving, recommendation sequence, distance, average gas consumption, and the number of recommended routes, ranking the top five of the feature importance. The following were subjective data, such as prior activities, the purposes of this trip, decision time, etc. However, no obvious relationships were found in demographic data, indicating that driver's route choice were less likely be affected by age and gender differences.

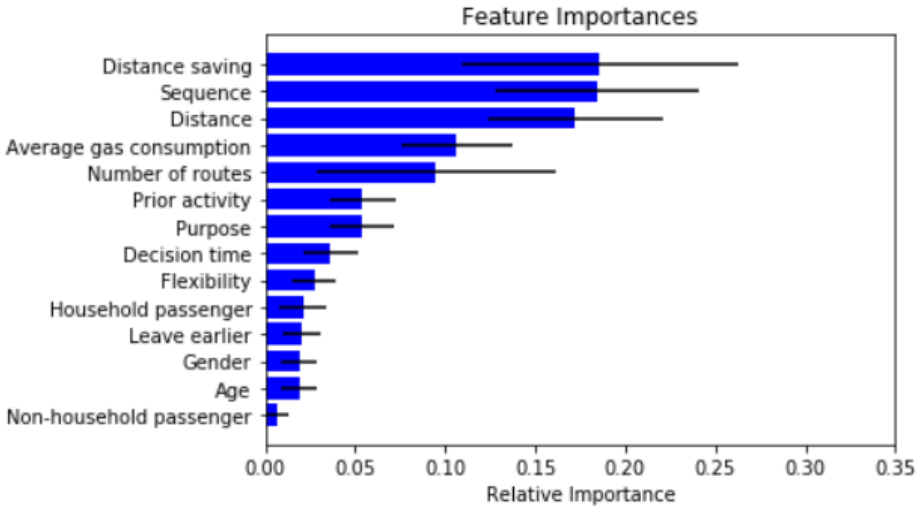


Figure 2.5. 13. Variable importance of the multi-label Random Forests

Impacting factors on following the recommended route

Generally, the average probability that drivers would actually follow the route after they chose from the recommended options was 56.7%, as shown in Figure 2.5.13. The detailed results of following probability when they selected different categories of recommended routes were illustrated in Figure 2.5.14. When drivers chose the eco, they had the largest likelihood (61.6%) to following the route, followed by selecting the fast one with an averaged probability 61.1%. When drivers selected the route with the balanced feature, they were the least likely to comply with their option (59.9%).

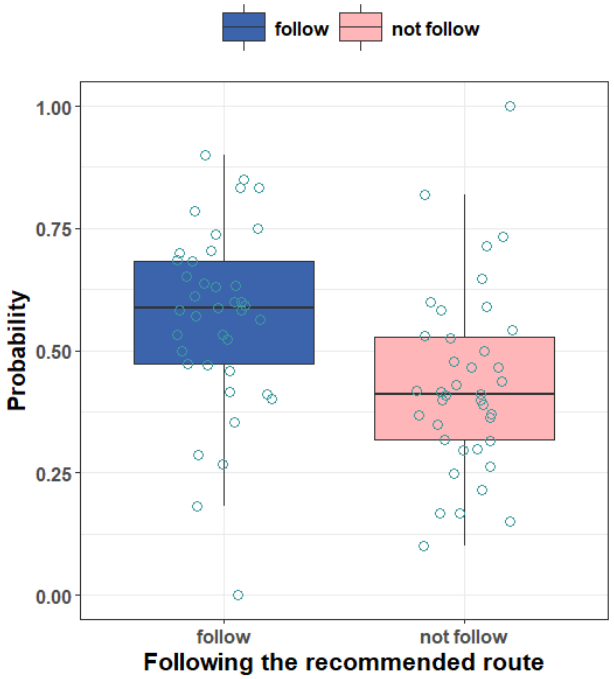


Figure 2.5. 14. The average probability of following the recommended route

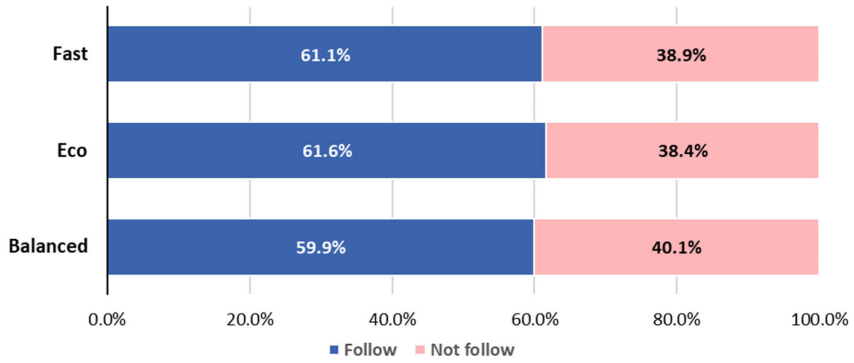


Figure 2.5. 15. The average following probability of each recommended route

To explore the impacting factors on following the recommended route, mixed model analyses were carried out by using the PROC GLIMMIX procedure. In total, 12 candidate variables were obtained from three sources:

- **Route information**: eco, fast, and balanced routes;
- **Driver characteristics**: age and gender;
- **Subjective data**: purpose, decision time, household passenger, non-household passenger, flexibility, prior activity, and leave earlier.

These variables together with interactions among them were treated as the fixed effects, while individual driver and interactions between driver and any fixed effects were regarded as random effects. Whether driver's following the selected route was the independent variable. Results were demonstrated in Table 2.5.5. If drive choose the eco ($t(32)=3.61$, $p=0.001$) or fast ($t(31)=4.68$, $p<0.001$) routes, they are more likely to fully drive along the recommended route. Additionally, compared with driving alone ($t(22)=-2.81$, $p=0.01$) or with only one household passenger ($t(22)=-2.95$, $p=0.007$), drivers will comply with the recommended route when there were three or more household passengers.

Table 2.5. 5 Mixed model results for the recommended route following

Effect	Estimate	Standard error	DF	t Value	Pr>|t|
Intercept	-1.199	0.257	22	-4.67	<0.001
Eco route	0.776	0.215	32	3.61	0.001
Fast route	1.071	0.229	31	4.68	<0.001
Household member					
0	-2.394	0.852	22	-2.81	0.010
1	-2.635	0.893	22	-2.95	0.007
2	-2.003	0.989	22	-2.02	0.055
3 or more*	0				

*Note: * denotes reference group for categorical variables; only significant factors were demonstrated in this Table.*

Conclusions

This study was conducted to collect naturalistic driving study data and understand what factors are impacting on drivers' decision making on route choices. A total of 738 valid trips from 39 participants were recorded and used in the final analysis. Overall, participants

followed the Recommended Route in 434 of 738 valid trips or 58.8 percent of the valid trips. Both Random Forest Tree algorithm and mixed models were applied in the analysis. In general, this study found that drivers would change their route choices under certain conditions, when they were provided with information related to different routes. Results of the analysis showed that drivers were more likely to select the eco route when its distance was shorter and gas consumption per mile was higher. It was also found that giving priority to recommend, the eco route could guide drivers to choose the eco way. If drive choose the eco or fast routes, they are more likely to fully drive along the recommended route while compared with driving alone or with only one household passenger, drivers will comply with the recommended route when there were three or more household passengers.

Task 3: Travel Behavior Modeling

The objective of task 3 is to study the impact of CAVs on travelers' travel behaviors, including departure time choice, route choice, willingness to ridesharing, etc.

Subtask	Content	Schedule	End date	Status
3.1	Experiment and survey design	M1	12/5/2015	Completed
3.2	Model departure-time choice behavior	M2-M6	5/5/2016	Completed
3.3	Model route choice behavior	M7-M12	11/5/2016	Completed
3.4	Model travel activity pattern change	M13-M20	7/5/2017	Completed
3.5	Calibration of POLARIS traveler behavior model	M21-M24	11/5/2017	Completed

Subtask 3.1. Experiment and survey design

A survey was designed to characterize the possible change of travel behaviors introduced by CAVs. The survey was conducted among the participants of the Safety Pilot project. In total, 396 responses were collected.

There are two types of questions in the questionnaire. The first type of the questions is related to the participants' demographic characteristics, such as gender, age, level of education, employment status, etc. The other type of questions is about the information of their households, for instance, number of cars, number of children, etc. Some of the questions are listed in Table 3.1.

Table 3.1 Sample questions in the survey

Questions	Values
Gender	1=Male, 2=Female
Age	1=25-34, 2=35-44, 3=45-54, 4=55-64, 5=65+
Primary driver of the vehicle	1=Yes, 2=No
Number of vehicles in the household	Integer
Number of adults in the household	Integer
Number of children in the household	1=0 children, 2=1 child, 3=2 children, 4=3 children, 5=4 children, 6=5+ children
Number of licensed drivers in the household	Integer
Status of Employment	1=Full time, 2=Part time, 3=No
Consistent work schedule	1=No, 2=Yes
Start time of work hours	Numerical
End time of work hours	Numerical

Primary mode of travel to work	1=Drive Alone, 2=Carpool, 3=AATA Bus, 4=U of M Bus, 5=Bicycle, 6=Walk, 7=Other
Status of student	1=Full time, 2=Part Time, 3=No
Highest level of education	2: Did not complete High School 3: High School/GED 4: Some college 5: Associate degree 6: Bachelor's Degree 7: Master's Degree
Bicycle	1=Yes, 2=No
Transit/bus pass	1=Yes, 2=No
Free parking at the place of employment	1=Yes, 2=No
Hours per week driving in Ann Arbor area	1=30 minutes or less, 2=1-4 hours, 3=5-9 hours, 4=10-14 hours, 5=15-20 hours, 6=Over 20 hours

This survey can be regarded as a benchmark when analyzing the possible change of travel behaviors. In subtask 3.3, when mining activity patterns, the data collected in this survey will be combined with the household activity data extracted from the Safety Pilot project database, to cluster the participants and analyze their similarities within each cluster.

Subtask 3.2. Departure time choice and traffic demand

3.2.1 Introduction

Traffic congestion and air pollution have become severe problems, especially in big cities. Ridesharing (Shaheen at al., 2015) is commonly recognized as an effective solution to reduce congestion and transport emission (Caulfield, 2009). Among all the factors that influence people's willingness to ridesharing, incompatible work schedules and loss of privacy are the top ones (Baldassare et al., 1998; Koppelman et al., 1993; Teal, 1987). Time savings from the usage of HOV lanes and monetary savings from the evenly shared tolls, parking fees, and gasoline costs are the main incentives to motivate people to share rides (Yang and Huang, 1999).

Autonomous vehicle (AV) technology can be a potential disrupter of the current mobility system. Since AV can reposition themselves, the negative effect of incompatible work schedules on ridesharing might be alleviated: when two commuters share a ride from home to attend two different activities, if their activity durations result in a long wait between the two when they would like to go home, an AV can be called to pick one of them up. However, with traditional vehicles, one has to either bear the long wait or use expensive taxi service. This observation implies that AV has a potential to encourage ridesharing when activity uncertainty is considered.

The goal of this subsection is to study the impact of AVs on travel mode and departure time choices during peak hours. We demonstrate the impact of AV in a simple scenario. The scenario considers round-trip commute with congestions at a single bottleneck and random work end time in the evening. In this scenario, a commuter can share a ride with another commuter from home to work in the morning, and either share a ride or call an AV to take her from workplace to home in the evening, depending on the actual work end time of her and her ridesharing partner. Dynamic user equilibria of the bottleneck congestion are analyzed for three cases: no ridesharing, ridesharing without AV, and ridesharing with AV.

3.2.2 Problem statement and classical bottleneck model

During the morning peak hours, in a lot of cities, for instance in New York, traffic flows mainly from outside of the city to the center of the city. Because of the huge traffic demand within a short period, congestion occurs easily. For simplicity, such transportation networks during the morning peak hours can be modeled as the graph shown on the left-hand of Figure 3.1. In the graph, "H" represents home (residential areas), and "W" represents workplace. Then, because all "H"-"W" pairs are independent from each other, we can just focus on one of the pairs, shown on the right-hand side of Figure 3.1.

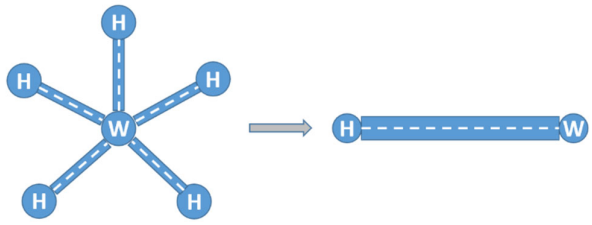


Figure 3.1 Simplification of transportation network with one center

In the standard bottleneck model proposed by Vikery (1969), commuters drive from "H" to "W" in the morning and wish to arrive at their workplace at a certain time point in the morning. "H" and "W" are connected by a road with capacity s. Since the capacity of the road is limited, once demand rate exceeds the capacity, there will be congestion. The commuters choose their departure time to minimize their travel costs. It is a tradeoff between suffering congestion and being early or late to arrive at workplace.

In the evening, similarly, most of the traffic flows from the city center to the outside. However, the difference is that commuters' departure time choice could be influenced by their work end time. Suppose two commuters go to work by ridesharing in the morning. If their work end time turns out to be very different, one of them must wait for the other for a long time. This effect of incompatible work schedule is one of the obstacles for ridesharing. To capture the effect of different work end time on commuters' travel behaviors, the work end time is modeled as a random variable distributed in an interval. Here provides a formal description of the model for the evening commute.

During the evening rush hours, a sufficiently large number, N, of commuters travel from work (W) to home (H). Each commuter can depart from (W) only after her work ends. The commuters' work end time is uniformly distributed in the time interval $[t_a, t_b]$, i.e., $t_e \sim \mathcal{U}(t_a, t_b)$. There is only one road connecting (H) and (W), with a single bottleneck whose capacity is s. If the arrival rate at the bottleneck exceeds s, a queue will form. If two commuters share a ride when they go to work in the morning, they should go home by ridesharing with each other as well. It's also assumed that ridesharing partners equally share their total travel costs.

Similar with the definitions in Vikery (1969) and Arnott et al. (1990), the cost of a trip, C, is expressed as a combination of the travel time spent on road, the travel time spent in office, and the fixed cost such as tolls. Specifically,

$$C = \text{travel time cost} + \text{waiting time cost} + \text{Fixed cost}$$

$$= \alpha(\text{travel time}) + \beta(\text{waiting time}) + \text{Fixed cost},$$

where α is the shadow value of the travel time when suffering from congestions on the road, and β is the shadow value of the time spent in the office after one's work ends. The fixed travel time cost is set to zero, which does not change the nature of the problem.

Considering both the realistic situations and the simplifications of the problem, the following assumptions are made.

- $\alpha > \beta$. Suffering from congestion on the road is costlier than waiting in the office.
- $\frac{N}{t_b - t_a} > s$. Peak demand density is larger than the capacity of bottleneck, otherwise no congestion will form, and each commuter will go home as soon as her work ends.
- Each commuter is rational and has complete information.

3.2.3 Round-trip bottleneck model without autonomous vehicles

To analyze the equilibrium of round-trip commute, we can study the behaviors of a pair of commuters in the following two cases.

Case 1: All the other commuters drive alone

Suppose all the other commuters share rides and then investigate the choice of a pair of commuters, by comparing the expected cost of ridesharing $\mathbb{E}(C_2)$ with the expected cost of driving alone $\mathbb{E}(C_1)$. If ridesharing is costlier, it is an equilibrium; if traveling alone is costlier, then it is not an equilibrium. In other words, an equilibrium obtains only if the two commuters have no incentive to share rides, i.e.,

$$\mathbb{E}(C_1) < \mathbb{E}(C_2).$$

Case 2: All the commuters share rides

Suppose all the other commuters share rides and then compare the costs for the two commuters when sharing a ride and when driving alone. If traveling alone is costlier, then an equilibrium obtains when all the people share rides. If ridesharing is costlier, then it is not an equilibrium. In other words, an equilibrium obtains only if the two commuters have no incentive to travel alone, i.e.,

$$\mathbb{E}(C_1) > \mathbb{E}(C_2).$$

3.2.4 Round-trip bottleneck model with autonomous vehicles

If all the regular vehicles are replaced by autonomous vehicles, the traffic pattern would be different. In the morning, all the commuters will go to work by ridesharing, because even if their work end time t_1, t_2 turn out to be very different from their partners', they can reposition an AV in advance to pick them up. The resultant total cost is still no more than the case when traveling alone during the round-trip commute.

In the evening, different from the case without autonomous vehicles, in this case, commuters can not only choose departure time but also decide if ridesharing or not. Their willingness to ridesharing will change the traffic demand and influence their departure time choice. On the contrary, their departure time choice can also change the traffic pattern and influence their willingness to ridesharing. Apparently, the fixed cost of a trip with an autonomous vehicle, F_{AV}, could change the equilibrium as well.

Due to the complexity of the problem, simulation-based method is applied. It basically simulates the process to achieve equilibrium. In each iteration, all the commuters choose their departure time and travel modes based on the observed traffic condition in the current iteration. Then, due to the choices made by the commuters, the traffic condition is also updated. After several iterations, commuters' choices and the corresponding traffic condition will converge, which means an equilibrium is achieved. Figure 3.2 shows the distribution of the travel modes in the (t_1, t_2) space, with $\beta/\alpha = 0.5, s(t_b - t_a)/N = 0.5, sF_{AV}/\alpha N = 0.05$. If the work end time of the two commuters lies in the green area, then they will still share a ride in the evening commute; otherwise, they will travel alone.

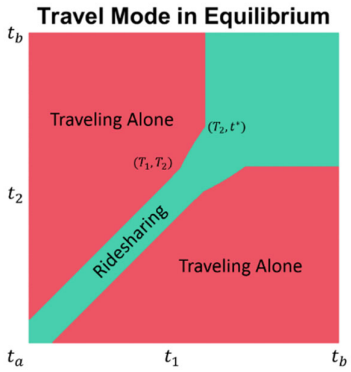


Figure 3.2 Travel mode in equilibrium considering AV

Figure 3.3 shows the percentage of ridesharing commuters during the evening commute for the three cases of equilibria when the vehicle related travel cost (fuel, mileage, tolls) varies. It can be easily seen that when the fixed cost is low, which our real-world situation usually is, AV can encourage more ridesharing. This is because with AV, commuters can go to work by ridesharing in the morning, without worrying about the potential incompatible work end time with their partners in the evening. In the evening, those pairs of commuters whose work end time turn out to be very close can still go home by ridesharing. However, when the fixed cost of an AV is high, AV can potentially discourage ridesharing. This is because with AV, those whose work end time is much earlier than their ridesharing partners can call AV to pick them up, without staying in the workplace and waiting for their partners.

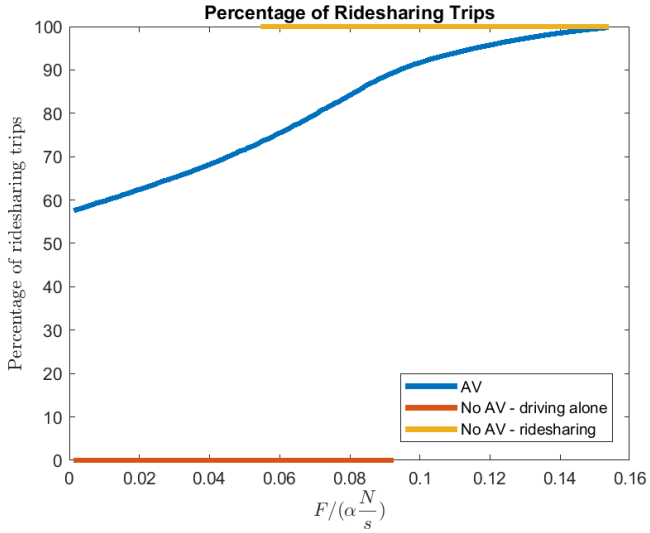


Figure 3.3 Percentage of ridesharing trips under different conditions

3.2.5 Summary

The objective of this study is to investigate the impact of autonomous vehicles on peak-hour traffic congestions. First, the congestion during peak hours is explained by bottleneck models. As an extension to the classical bottleneck model which focuses on morning commute only, the methodology is also applied to the evening trip and thus the round-trip commute. Then, the equilibria of departure time and travel mode choices are established for both regular vehicle case and autonomous vehicle case. It is shown that in equilibria, AV could encourage ridesharing if its fixed cost is low enough but discourage ridesharing if its cost is high.

It should also be noted that this study has limitations that future work may overcome. On one hand, in this study, a simplified transportation network is used. In real world, a transportation system can be very large and complicated. It may also contain HOV lanes, public transportation, parking constraints, etc. On the other hand, all the commuters in this study are assumed to be rational and homogeneous. In real life, commuters may have bounded rationality and different values of time.

Subtask 3.3-3.5. Activity patterns and travel behaviors

3.3.1 Activity pattern mining

Introduction

To understand the potential changes of departure-time choices, route choices, and travel activity patterns after CAVs are in place, the first step is to understand how people travel now when CAVs are not prevalent. To this end, analyzing the baseline travel pattern using Safety Pilot data is essential. At this stage, we hope to travel patterns (including departure-time, route choice, activities, and destinations) from the Safety Pilot database, conducting statistical analysis and build a connection between the trajectory data and the travelers' demographic factors.

Traveling is a derived demand and the internal motivation for traveling is to accomplish certain activities. Therefore, the activity-based approach provides a more realistic and enriched representation of how people travel compared to the traditional trip-based approach. With the emergence of the trajectory data collection devices, such as GPS devices, smart phones, Bluetooth, or connected vehicles, the activity-based approach becomes increasingly popular in travel demand modeling and forecasting because of its power in accommodating these disaggregated travel data.

Data description

Specifically, we will introduce the data collected from the first nationwide scaled-down testbed for connected vehicles in Ann Arbor, Michigan. "Stay points" along a GPS trace (i.e., a static point with a staying duration of more than a certain threshold) contains importation travel "semantic" information, including the type of the land use for each stay point, the activity one performs at that location, and the activity schedule for each individual traveler. Such rich information can benefit various research topics, including: activity travel pattern identification, hotspot analysis (i.e., a region with concentrated travel demands), travel recommendation, life pattern understanding, user similarity, and location prediction (Zheng, 2015).

One trip is defined as one segment of travel starting from when the engine is turned on until it is off. Literally, the first GPS point along one trip is the origin while the last is the destination (see Figure 3.4). If a driver is assumed not to stop in the middle of a trip, the origins and destinations are treated as the major stay points for each individual.

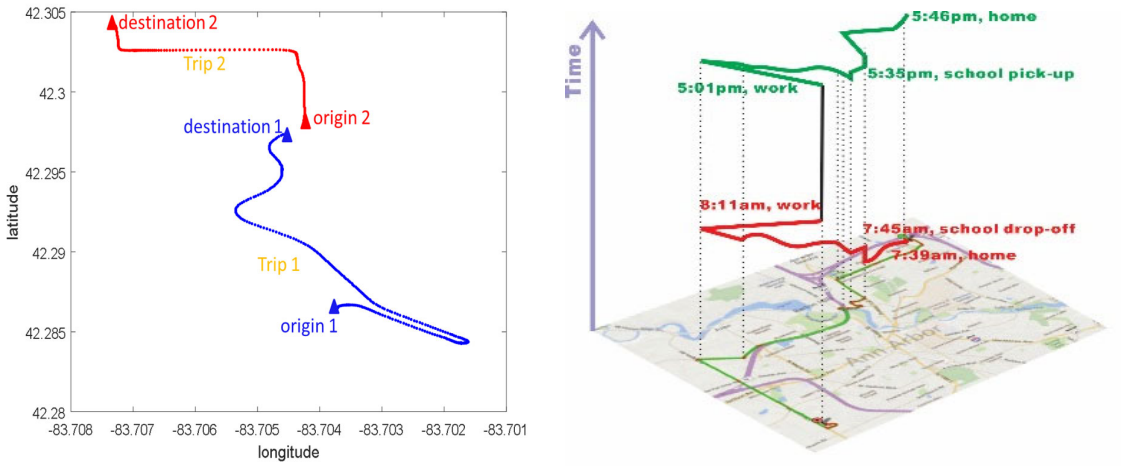


Figure 3.4 Sample travel-activity pattern: (a) Conceptual trip chain; (b)Time-space trajectory

However, GPS trajectories are generally quite noisy and thus in many cases, we cannot directly adopt the first (or last) points of a trip as the origins (or destinations). Especially when the car engine just starts, it takes a while for GPS devices to connect to satellites and thus make origin points particularly difficult to identify. To more accurately capture origins and destinations, we used both speed, denoted by v, and the number of satellite, denoted by n. The rationale of using speed is that when a trip just starts or is about to end, its speed must be lower than a threshold speed v_h. The number of satellites attached to one GPS point indicates its measurement accuracy. The more satellites, the more accurate the point is. In other words, assume the optimal number of satellites to ensure a GPS measure is n_s, the closer the actual satellite number is from n_s, the more accurate this point is. Accordingly, the confidence of the accuracy of a GPS point is defined as the weighted average of the speed difference from the threshold speed and the satellite number difference from the optimal number:

$$Q = \alpha\left(\frac{v_h - v}{v_h}\right) + \beta\left(\frac{n_s - n}{n_s}\right)$$

where α and β are weight parameters.

Built upon the confidence of a point, now we are ready to define a set of rules to identify the origin $O(i)$ and the destination $D(i)$ for a trip T_i. If the confidence of the first (or last) point is greater than 0.6, it will be used as the origin (or destination) point. However, if its confidence is lower than 0.6, it is inappropriate to use. Given that an individual's travel trajectories are continuous, the origin of the trip T_i is the destination of the trip T_{i-1} and the destination of the trip T_i is the origin of the trip T_{i+1}. Therefore, if the confidence of the destination (or origin) of the last (or next) trip is higher than the first (or last) point of this trip, we will use the destination (or origin) of the last (or next) trip. Otherwise we will not consider this trip. These rules can be expressed as follows:

$$O(i) = \begin{cases} first(T_i) & Q\big(first(T_i)\big) \geq 0.6 \\ D(i-1) & Q\big(D(i-1)\big) \geq Q\big(first(T_i)\big) \;\&\; Q\big(first(T_i)\big) < 0.6, \\ \emptyset & other \end{cases}$$

$$D(i) = \begin{cases} last(T_i) & Q\big(last(T_i)\big) \geq 0.6 \\ O(i+1) & Q\big(O(i+1)\big) \geq Q\big(last(T_i)\big) \;\&\; Q\big(last(T_i)\big) < 0.6, \\ \emptyset & other \end{cases}$$

where first(T_i) represents the first point of trip T_i; last(T_i) represents the last point of trip T_i; $D(i-1)$ and $O(i+1)$ represent destination of the trip T_{i-1} and the origin of the trip T_{i+1} respectively.

Using the above rules, all the stay points of each driver can be extracted. Here the data used is from April 1 to October 30, 2013. Figure 3.5 illustrates one driver's stay points during this period where x-axis and y-axis represent longitude and latitude respectively. Clearly these points are highly scattered, mainly because of the GPS errors or because drivers may stop in multiple places in one mall area. DBSCAN (Density-based spatial clustering of applications with noise) method (Ester et al., 1996) is first applied in MATLAB if points are within the range of 50 meters. As illustrated, the blue cross in the circle is the center of a cluster of multiple points. If a point is visited at least once a week, it is a frequently visited place (in blue), otherwise it is an infrequently visited place (in red).

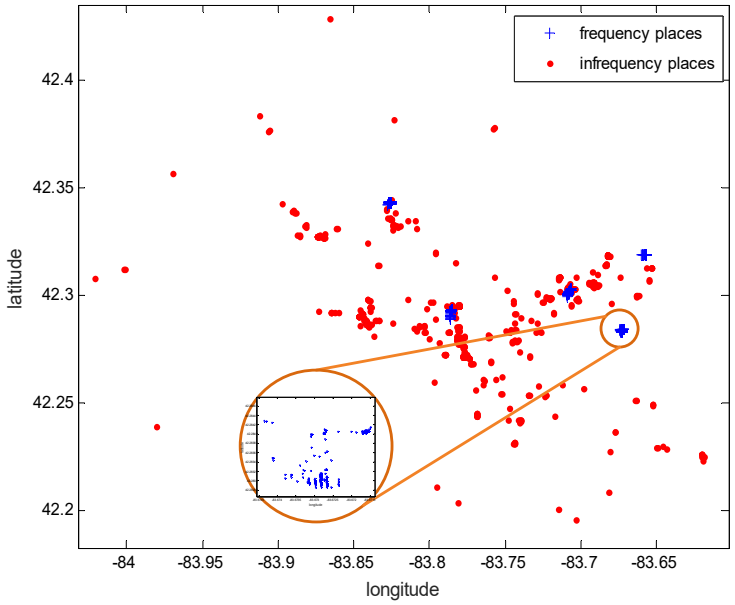


Figure 3.5 Density cluster of the origins and the destinations for one driver

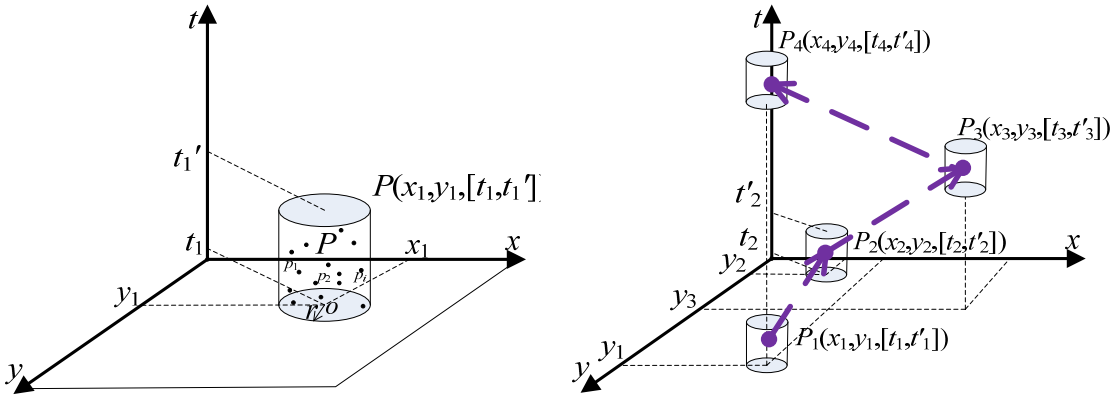
To identify the land use types of these points, they are then imported to ArcGIS (www.argcis.com) and the "spatial union" tool is used to map points to the shape file of the City of Ann Arbor land use with the threshold value set as 100 meters. There are 13 land use types, as shown in Table 3.2.

Table 3.2 Land use types

Index	Lane use type
0	On road
1	Single-family/ Multiple-family residential
2	Commercial
3	Industrial
4	Governmental / Institutional
5	Parks, Recreation, and Open Space
6	Transportation/communication/utilities (TCU)
7	Water
8	Agricultural
9	Airport
10	Others
11	Home

12	Work

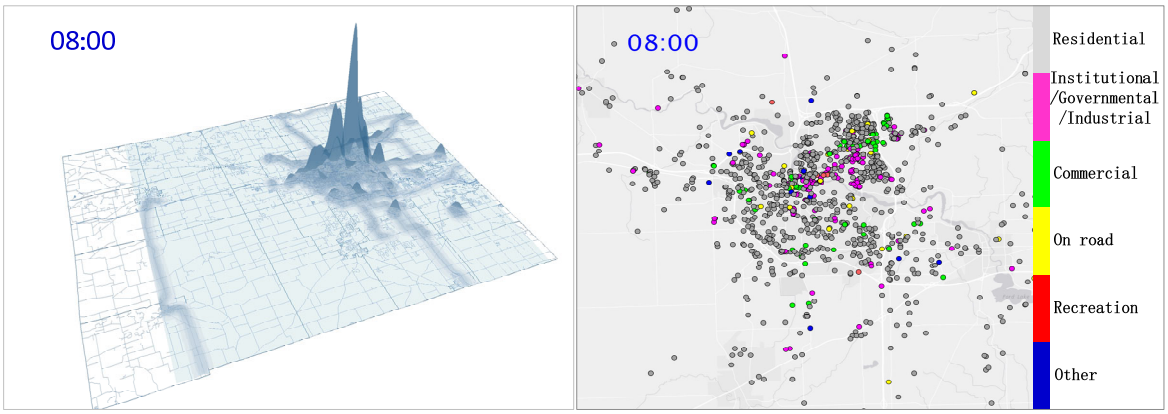
In a period of 7-month travel history, a set of time stamps to visit one cluster can be extracted. In Figure 3.6, the x-axis and y-axis are longitude and latitude respectively while z-axis represents the time stamp. Denote the coordinate of the cluster center by (x_1, y_1). Denote the earliest time to arrive at one cluster and the latest time to leave the cluster by t_1 and t_1' respectively. Therefore, one frequently visited place can be denoted by $P(x_1, y_1, [t_1, t_1'])$. A traveler's activity pattern ($P1 \rightarrow P2 \rightarrow P3 \rightarrow P4$) is obtained when time $t_{i+1} > t_i'(i = 1,2, \dots)$, where t_{i+1} is the earliest time to reach the activity $i+1$ and t_i' is the latest time to leave activity i.

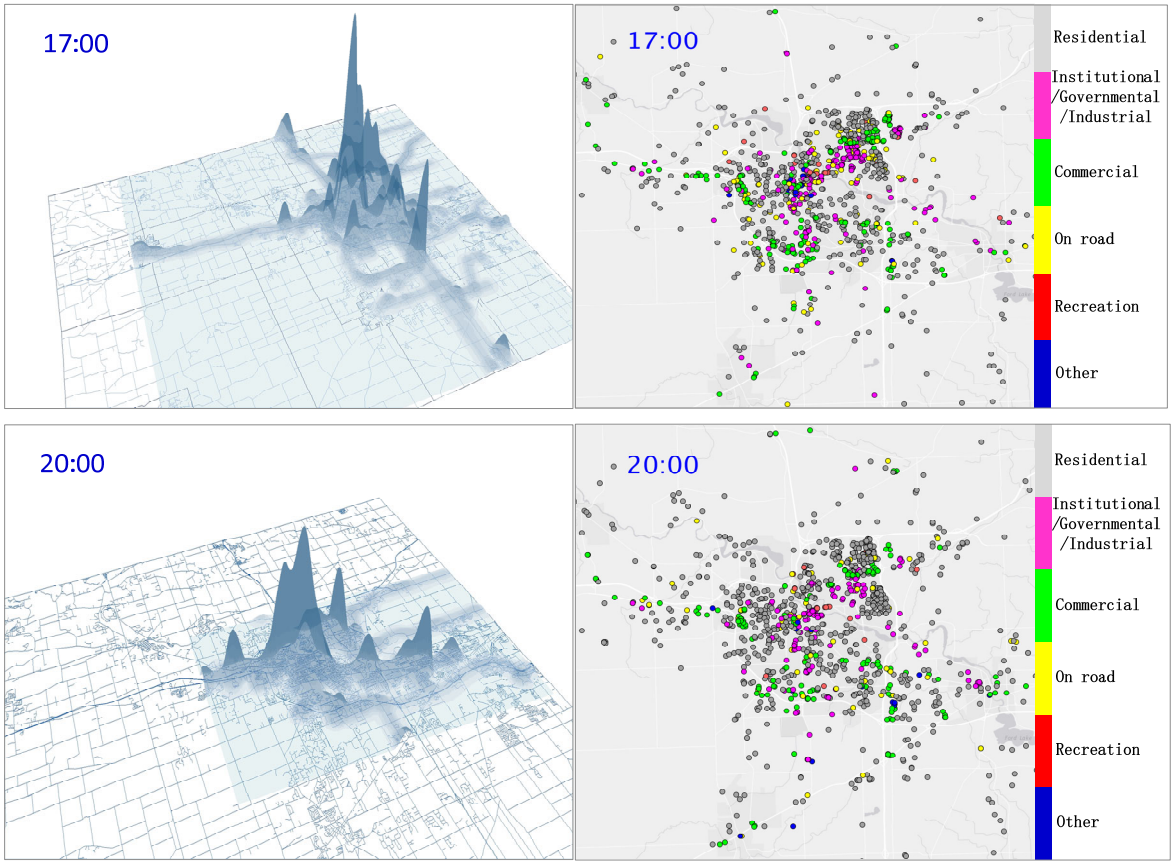


(a) A cluster of frequently visited places (b) Activity chain

Figure 3.6 Schematic diagram of frequently visited places and activity pattern

To understand the evolution of the activity patterns, the dynamic activity patterns are presented over the map of the city of Ann Arbor on Friday September 13, 2013 using a professional data visualization tool "Processing" (www.processing.org). Figure 3.7 shows the traffic density during a one-hour period (on the left column) and the activity type of each participant's residing location at a given time (on the right column). As time elapses, the cars start leaving residential areas and stay in the land use types related to work from 8 am. After 5 pm, more cars are on road during the evening peak hour. At 8 pm, there is a growing number of cars on commercial or recreational land use.





(a) Traffic volume (b) Activity patterns

Figure 3.7 Schematic diagram of frequently visited places and activity pattern

Similarity analysis among activity patterns

Similarity is a quantitative measure of the extent to which patterns from two sources are alike. Cosine similarity, measuring the cosine of the angle between two vectors, is widely used in revealing connections in the social network (Toole et al., 2015).

In this study, we aim to define a similarity index measuring individual's activity sequence similarity. Travel activity pattern similarity is a quantitative measure of the extent to which travel activity patterns of two households or individuals are alike. To find an appropriate similarity measure between two activity patterns, we first define the longest common sequence between two sequences a_i and a_j. For instance, if individual i's pattern is 'home-work-institution-home' and individual j's is 'home-work-shops-work-home', then their longest common sequence is 'home-work-home'. Accordingly, the travel pattern similarity is defined as:

$$s_{ij} = \frac{length\big(longest\ common\ sequence\ between\ a_i\ and\ a_j\big)}{\max\{length(a_i), length\big(a_j\big)\}}$$

Note that s_{ij} is always equal to s_{ji} so the similarity matrix is symmetric. This definition does not consider where and when each activity is conducted but only the sequence.

Built upon the similarity calculated between every two travelers, we will group them by their similarity values and then compare their demographic features within each cluster. The similarity value s_{ij} for individual i and j can be used to construct a similarity graph, denoted as G, which is composed of nodes and edges $G = (V, E)$, where V is the set of households and E is the set of weighted edges. The

weight of the edge ij is s_{ij}, representing the similarity between i and j. Spectral clustering method is implemented to cluster travelers sharing similar travel patterns (Von Luxburg, 2007).

Applying the clustering method to the Safety Pilot dataset, 3 major clusters are identified. Looking into the demographic characteristics, significant difference among the clusters can be observed, as shown in Figure 3.8. For instance, people in cluster 1 spent a lot of time at school. In fact, all the travelers in cluster 1 have child/children in their households. In cluster 2, travelers spent more time at workplace. In fact, all of them are employed, as the survey tells. People in Cluster 3 spent more time at home. In fact, they are mostly senior people according to the survey.

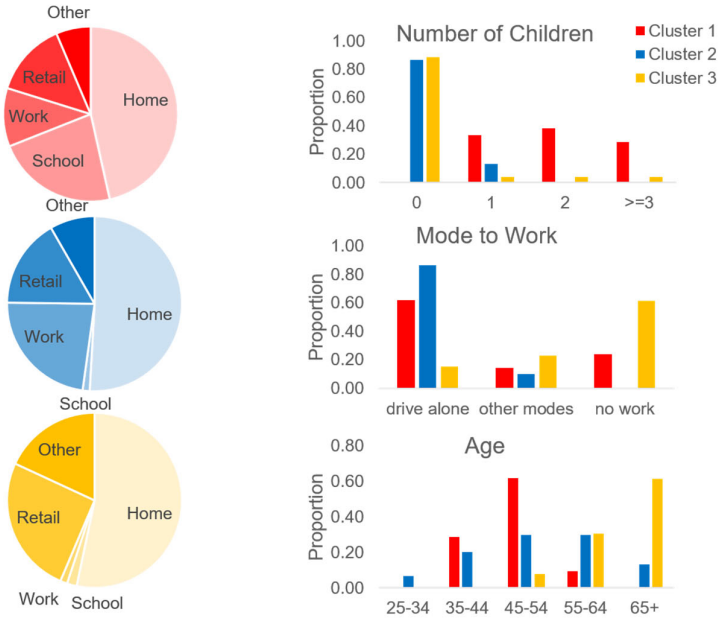


Figure 3.8 Histogram of household features within five clusters

Other than inspecting features within each cluster using histograms, it is also interesting to know whether travel activity similarity is correlated with any demographic features quantitatively. There are many machine learning techniques which can be used to establish the mapping from demographic features to travel behavioral similarity. For simplicity, the linear regression model, i.e., $S = \phi X + b$, is used. As S, X are matrices, they are first vectorized and then the regression is performed in the corresponding vector space. The elements in X are categorical variables, so they are first converted to factors and then multiple linear regression is employed in R using "lm" routine. Table 3.3 illustrates the estimation results using 396 samples.

Table 3.3 Estimation of coefficients

	Estimate	Std. Error	t value	p value	Significance level
Intercept	0.615548	0.002234	275.516	< 2e-16	***
Gender2	-0.00509	0.001303	-3.911	9.21E-05	***
Age2	-0.00176	0.001635	-1.077	0.2814	
Age3	0.00329	0.001856	1.773	0.07627	.
Age4	0.01238	0.002554	4.848	1.25E-06	***
Age5	-0.01552	0.005181	-2.995	0.00274	**
Adults2	0.003693	0.001499	2.463	0.01377	*
Adults3	0.006318	0.002273	2.78	0.00544	**
Adults4	0.015908	0.003711	4.287	1.81E-05	***

Adults5	0.036668	0.008661	4.234	2.30E-05	***
Children2	-0.01623	0.001539	-10.546	< 2e-16	***
Children3	-0.04346	0.001802	-24.116	< 2e-16	***
Children4	-0.03711	0.002717	-13.66	< 2e-16	***
Children5	-0.03272	0.010595	-3.088	0.00201	**
Children6	-0.01508	0.006015	-2.508	0.01215	*
Vehicles2	-0.00846	0.001539	-5.493	3.95E-08	***
Vehicles3	-0.02018	0.002138	-9.442	< 2e-16	***
Vehicles4	-0.00926	0.003455	-2.68	0.00737	**
Vehicles5	0.012982	0.00584	2.223	0.02622	*
Vehicles6	-0.00338	0.012988	-0.261	0.79442	
Employed2	-0.0319	0.001502	-21.236	< 2e-16	***
Schedule2	-0.02041	0.001349	-15.134	< 2e-16	***
Mode2	-0.02563	0.002999	-8.544	< 2e-16	***
Mode3	0.016646	0.002921	5.698	1.21E-08	
Mode4	-0.00146	0.007034	-0.208	0.83522	
Mode5	-0.00054	0.003439	-0.156	0.87572	
Mode6	-0.05617	0.002975	-18.879	< 2e-16	***
Mode7	-0.07191	0.002231	-32.227	< 2e-16	***
Student2	-0.02562	0.002804	-9.138	< 2e-16	***
Student3	-0.06731	0.003754	-17.93	< 2e-16	***
Education3	-0.00452	0.001529	-2.958	0.0031	**
Education4	-0.00386	0.001924	-2.007	0.04474	*
Education5	-0.00852	0.002772	-3.075	0.0021	**
Education6	0.005668	0.00439	1.291	0.19665	
Education7	-0.0299	0.014079	-2.124	0.03371	*

Significance codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Because all the explanatory variables are categorical factors, the first level is used as the reference level. The number after the variable name represents each level other than the reference level. For example, gender has two levels: 1 (male) and 2 (female). In R, gender=1 is treated as the reference level and therefore the coefficient (i.e., -0.00509) in front of "Gender2" is estimated relative to the reference level, meaning that when the gender is changed from 1 to 2, the similarity measure will be reduced by a magnitude of 0.00509. Most coefficients are significant because of the small p-value, indicating that an individual's travel activity is indeed correlated to demographic features.

3.3.2 Household activity pattern optimization model

Introduction

The goal of this subsection is to investigate the impact of automated vehicles on daily traffic and energy consumption. The model is activity-based because the presence of automated vehicles will not only change people's travel experience on road, but also change their departure time and travel mode. To this end, we should change our focus from how people finish their daily travels to how people finish their daily activities.

The household activity pattern optimization problem is defined as follows. For a household that has M members and V vehicles, given the daily activity of each member, determine the optimal activity sequence for each member/vehicle as well as the start and end time of each activity, so that all activities can be finished, and the system cost is minimized. The system cost could be one or a

combination of the followings: total travel time, total waiting time, total travel distance, total energy consumption and so on. There can also be some constraints, such as

- Travel time between different locations of activities
- Feasible time windows for start/end of each activity
- Some activities must be performed by certain members
- Some activities must be performed using certain vehicles
- Some members can only drive certain vehicles

Optimization model

The problem is formulated using a network flow model. The start or the end of each activity is modeled as a node, and connections between each pair of nodes are the links. Note that the links in this model have two categories. If the link is from the start of an activity to the end of the same activity, the weight on the link is the duration of the activity; if the link is from the start/end of one activity to the start/end of another activity, the weight on the link is the travel time between two locations. With these nodes and links, the goal of the problem is to find an optimal path for each member and each vehicle to finish all activities with minimal cost. The notations are summarized in Table 3.4.

Table 3.4 Notations used in the household activity pattern optimization model

Input parameters	
AV	The set of indicators whether each vehicle is an autonomous vehicle. **AV**(v)=1, if vehicle v is an automated vehicle. **AV**(v)=0, otherwise.
RS	The matrix of indicators whether a member is willing to share a ride with another member. **RS**(m,n)=1, if member m want to share ride with member n. **RS**(m,n)=0, otherwise.
MA	Member-activity matrix. **MA**(m,a)=1, if member m is eligible to perform activity a. **MA**(m,a)=0, if member m is not eligible to perform activity a.
VA	Vehicle-activity matrix. **VA**(v,a)=1, if vehicle v can be used to perform activity a. **VA**(v,a)=0, otherwise.
$\mathbf{MV_c}$	Member-vehicle matrix of capability. $\mathbf{MV_c}(m,v)$=1, if member m can drive vehicle v. $\mathbf{MA_c}(m,v)$=0, otherwise.
$\mathbf{MV_o}$	Member-vehicle matrix of ownership. $\mathbf{MV_o}(m,v)$=1, if member m own vehicle v. $\mathbf{MA_o}(m,v)$=0, otherwise. Here we assume that each vehicle must stay with one of its owners at the beginning and the end of the day.
VC	The set of vehicle capacities. **VC**(v) denotes the capacity of vehicle v.
$\mathbf{P}^+$	The set of spatial-temporal nodes corresponding to arriving at the locations of activities. $\mathbf{P}^+(a)$ denotes arriving at the location of activity a.
$\mathbf{P}^-$	The set of spatial-temporal nodes corresponding to leaving from the locations of activities. $\mathbf{P}^-(a)$ denotes leaving from the location of activity a.
FP	For any activity a, if $u=\mathbf{P}^+(a)$, then $\mathbf{FP}(u)=\mathbf{P}^-(a)$.
P	$\mathbf{P}=\mathbf{P}^+\cup\mathbf{P}^-$.
$\mathbf{Q}^+$	The set of spatial-temporal nodes corresponding to returning home, including the final return home and midday return home. $\mathbf{Q}^+(m,k)$ denotes the k^{th} return home for member m, k=1,2,…, K, K+1, where K is the maximum number of midday return home, $\mathbf{Q}^+(m, K+1)$ denotes the final return home for member m.
$\mathbf{Q}^-$	The set of spatial-temporal nodes corresponding to midday leaving home, including the first departure from home and midday departure from home. $\mathbf{Q}^-(m,k)$ denotes the k^{th} leaving home for member m, k=0,1,2,…, K, where K is the maximum number of midday leaving home, $\mathbf{Q}^-(m,0)$ denotes the first departure from home for member m.
N	The set of all spatial-temporal nodes. $N=\{\mathbf{P}^+, \mathbf{P}^-, \mathbf{Q}^+, \mathbf{Q}^-\}$.

t	Time matrix. $t_{u,w}$ is a large number L if w=u or u=**FP**(w), or the activity execution time if w=**FP**(u), or the travel time between the location of node u to location of node w for all other cases.
c	Cost matrix. $c^v_{u,w}$ denotes the vehicle cost (e.g., fuel consumption) if vehicle v is used to travel from node u to node w.
TW	The matrix of time window for all the nodes including the first departure and the final return. Specifically, **TW**=[$\mathbf{TW}_a$ $\mathbf{TW}_b$], where [$\mathbf{TW}_a$(u) $\mathbf{TW}_b$(u)] gives the time window for activity u.
Decision variables	
$X^m_{u,w}$	Binary variable indicating whether link (u,w) from node u to node w is on the path of member m. $X^m_{u,w}$=1, if link (u,w) from node u to node w is on the path of member m. $X^m_{u,w}$=0, otherwise.
$X^v_{u,w}$	Binary variable indicating whether link (u,w) from node u to node w is on the path of vehicle v. $X^v_{u,w}$=1, if link (u,w) from node u to node w is on the path of vehicle v. $X^v_{u,w}$=0, otherwise.
T_u	Continuous variable denoting the time of each spatial-temporal node.

The constraints in this problem include:

1) Each activity must be performed by at least one eligible member.
$$\sum_{m\in\mathbf{M}}\left(X^m_{u,\mathbf{FP}(u)}\cdot\mathbf{MA}(m,u)\right)\geq 1,\ u\in\mathbf{P}^+$$

2) Each activity must be performed by an eligible vehicle.
$$\sum_{v\in\mathbf{V}}\left(\sum_{w\in\mathbf{N}}X^v_{w,u}\cdot\mathbf{VA}(v,u)\right)=1,\ u\in\mathbf{P}^+$$

$$\sum_{v\in\mathbf{V}}\left(\sum_{w\in\mathbf{N}}X^v_{u,w}\cdot\mathbf{VA}(v,u)\right)=1,\ u\in\mathbf{P}^-$$

3) Midday return home trips may or may not happen for each member or vehicle.
The number of midday return home trips for each member or vehicle is restricted to a certain maximum value, K. One obvious upper bound is the total number of locations for all activities and homes, N_A+N_M. However, it is usually much lower than that.

4) Flow conservation for each member.
For the ease of composition, we assume there is always a dummy link between node $\mathbf{Q}^+$(m,K+1) and node $\mathbf{Q}^-$(m,0) for member m, i.e., $X^m_{u,w}$=1, if u=$\mathbf{Q}^+$(m,K+1) and w=$\mathbf{Q}^-$(m,0). The physical meaning of this link is that member m should stay at his/her home from the final return home time on one day to the first departure time on the following day. With that, we can complete the following conservation equation for member m.
$$\sum_{w\in\mathbf{N}}X^m_{u,w}=\sum_{w\in\mathbf{N}}X^m_{w,u},\ m\in\mathbf{M},\ u\in\mathbf{N}$$

5) Flow conservation for each vehicle.
Similarly, assume that $X^v_{u,w}$=1, if u=$\mathbf{Q}^+$(m,K+1), w=$\mathbf{Q}^-$(m,0) and $\mathbf{MV}_o$(m,v)=1, for any vehicle v and one of its owner m. The flow conservation equation for vehicle v is
$$\sum_{w\in\mathbf{N}}X^v_{u,w}=\sum_{w\in\mathbf{N}}X^v_{w,u},\ v\in\mathbf{V},\ u\in\mathbf{N}$$

6) Each node should be visited at most once for any member or vehicle.

$$\sum_{w\in\mathbf{N}} X^m_{u,w} \le 1,\ m\in\mathbf{M}, u\in\mathbf{N}$$

$$\sum_{w\in\mathbf{N}} X^v_{u,w} \le 1,\ v\in\mathbf{V},\ u\in\mathbf{N}$$

7) Driver on board.
For any trip of a regular vehicle *v*, it must be guaranteed that at least one feasible driver of that vehicle is on board. However, if vehicle *v* itself is an automated vehicle, there would be no constrained on human drivers.

$$\sum_{m\in\mathbf{M}} \left(X^m_{u,w}\cdot \mathbf{MV}(m,v)\right) + \mathbf{AV}(v) \ge 1, \text{if } X^v_{u,w} = 1,\ v\in\mathbf{V}, u,w\in\mathbf{N}$$

8) Travel with vehicle.
For any link that is part of any member's path, it must be guaranteed that there is one vehicle traveling on the same link.

$$\sum_{v\in\mathbf{V}} X^v_{u,w} = 1, \text{if } \sum_{m\in\mathbf{M}} X^m_{u,w} \ge 1,\ u,w\in\mathbf{N}$$

Note that there is no need to let two vehicles travel on the same link, because each node in the model is specific for one member.

9) Vehicle capacity constraint.
The number of occupants of any vehicle at any time must be less than or equal to its capacity.

$$\sum_{m\in\mathbf{M}} X^m_{u,w} \le \mathbf{VC}(v), \text{if } X^v_{u,w} = 1,\ v\in\mathbf{V}, u,w\in\mathbf{N}$$

10) Ride-sharing.
Each member *m* will not visit any activity that is not on his/her list. However, if member *m* wants to share a ride with another member *n*, then member *m* may also visit member *n*'s activities and member *n*'s home.

$$\sum_{w\in\mathbf{N}} \left(X^m_{u,w} + X^m_{\mathbf{FP}(u),w} + X^m_{w,u} + X^m_{w,\mathbf{FP}(u)}\right) = 0,\ \text{if } \sum_{n\in\mathbf{M}} \mathbf{RS}(m,n)\cdot\mathbf{MA}(n,u) = 0,$$

$$\text{or, } \sum_{w\in\mathbf{N}} \left(X^m_{u,w} + X^m_{\mathbf{FP}(u),w} + X^m_{w,u} + X^m_{w,\mathbf{FP}(u)}\right) \le L\cdot \sum_{n\in\mathbf{M}} \mathbf{RS}(m,n)\cdot\mathbf{MA}(n,u)$$

$$m\in\mathbf{M}, u\in\mathbf{P}^+$$

$$\sum_{u\in\mathbf{N}} \left(X^m_{u,h^+_n} + X^m_{u,h^-_n} + X^m_{h^+_n,u} + X^m_{h^-_n,u}\right) = 0,\ \text{if } \mathbf{RS}(m,n) = 0,$$

$$\text{or, } \sum_{u\in\mathbf{N}} \left(X^m_{u,h^+_n} + X^m_{u,h^-_n} + X^m_{h^+_n,u} + X^m_{h^-_n,u}\right) \le L\cdot \mathbf{RS}(m,n)$$

$$m,n\in\mathbf{M}, m\ne n$$

Similarly, each vehicle v will not visit any activity or home location that is not on the list of any of its owners or its owners' ride-mates.

$$\sum_{w\in\mathbf{N}}\left(X^v_{u,w}+X^v_{\mathbf{FP}(u),w}+X^v_{w,u}+X^v_{w,\mathbf{FP}(u)}\right)=0,$$

$$\text{if } \sum_{m\in\mathbf{M}}\left(\mathbf{MV_0}(m,v)\cdot\sum_{n\in\mathbf{M}}\left(\mathbf{RS}(m,n)\mathbf{MA}(n,u)\right)\right)=0, v\in\mathbf{V}, u\in\mathbf{P}^+$$

$$\text{or, } \sum_{w\in\mathbf{N}}\left(X^v_{u,w}+X^v_{\mathbf{FP}(u),w}+X^v_{w,u}+X^v_{w,\mathbf{FP}(u)}\right)\le$$

$$L\cdot\sum_{m\in\mathbf{M}}\left(\mathbf{MV_0}(m,v)\cdot\sum_{n\in\mathbf{M}}\left(\mathbf{RS}(m,n)\mathbf{MA}(n,u)\right)\right)$$

$$\sum_{u\in\mathbf{N}}\left(X^v_{u,h_n^+}+X^v_{u,h_n^-}+X^v_{h_n^+,u}+X^v_{h_n^-,u}\right)=0,\ \text{if } \sum_{m\in\mathbf{M}}\left(\mathbf{RS}(n,m)\mathbf{MV_0}(m,v)\right)=0,$$

$$\text{or, } \sum_{u\in\mathbf{N}}\left(X^v_{u,h_n^+}+X^v_{u,h_n^-}+X^v_{h_n^+,u}+X^v_{h_n^-,u}\right)\le L\cdot\sum_{m\in\mathbf{M}}\left(\mathbf{RS}(n,m)\mathbf{MV_0}(m,v)\right)$$

$$n\in\mathbf{M}, v\in\mathbf{V}$$

11) Time constraints.

For any node, its time must be within the feasible time window.

$$\mathbf{TW}_a(u)\le T_u\le\mathbf{TW}_b(u), u\in\mathbf{N}$$

For any link connecting two consecutive spatial-temporal nodes, except for the dummy link, the time difference between them must be greater or equal to the time needed to finish the trip or activity.

$$T_w-T_u\ge t_{u,w}, \text{if } X^m_{u,w}=1 \text{ or } X^v_{u,w}=1,\ m\in\mathbf{M},\ v\in\mathbf{V},\ u,w\in\mathbf{N},\ w\ne\mathbf{Q}^-(m,0),$$

where $t_{u,w}$ is a large number L if $w=u$ or $u=FP(w)$, or the activity execution time if $w=FP(u)$, or the travel time between the location of node u to location of node w for all other cases.

Cost functions could include one or more of the following:

Cost function	Description
$\sum_{v\in\mathbf{V}}\sum_{u\in\mathbf{N}}\sum_{w\in\mathbf{N}}c^v_{u,w}X^v_{u,w}$	Total cost of vehicle usage.
$\sum_{m\in\mathbf{M}}\sum_{u\in\mathbf{N}}\sum_{w\in\mathbf{N},w\ne\mathbf{FP}(u)}t_{u,w}X^m_{u,w}$	Total travel time for all members.
$\sum_{m\in\mathbf{M}}T_{\mathbf{Q}^+(m,K+1)}$	Final return home time.
$\sum_{m\in\mathbf{M}}\left(T_{\mathbf{Q}^+(m,K+1)}-T_{\mathbf{Q}^-(m,0)}\right)$	Total out-of-home time
$\sum_{u\in\mathbf{P}^+}\left(\sum_{m\in\mathbf{M}}X^m_{u,\mathbf{FP}(u)}(T_{\mathbf{FP}(u)}-T_u)-t_{u,\mathbf{FP}(u)}\right)$	Total waiting time.
$\sum_{u\in\mathbf{P}^+}\left(T_{\mathbf{FP}(u)}-T_u-t_{u,\mathbf{FP}(u)}+\left(\sum_{m\in\mathbf{M}}X^m_{u,\mathbf{FP}(u)}-1\right)\cdot t_{u,\mathbf{FP}(u)}\right)$	Approximated total waiting time*
$\sum_{m\in\mathbf{M}}\left(T_{\mathbf{Q}^+(m,K+1)}-T_{\mathbf{Q}^-(m,0)}\right)-\sum_{m\in\mathbf{M}}\sum_{u\in\mathbf{N}}\sum_{w\in\mathbf{N},w\ne\mathbf{FP}(u)}t_{u,w}X^m_{u,w}$	Total waiting time (alternative expression**)

* Here we approximate the waiting time for any additional member engaged in an activity as the activity execution time. When this approximate total waiting time is minimized, it is very likely that the actual total waiting time is also minimized.

** Note that the out-of-home time for all members is comprised of the total travel time for all members, the total execution time for all activities (a constant), and the total waiting time for all members.

3.3.3 Minimum fleet problem

Problem statement

If multiple families share a fleet of autonomous vehicles, what is the minimum fleet size required to satisfy their travel demand? Specifically, given a set of origins and destinations of all the trips, how many vehicles are needed to serve the trips? How will the vehicle miles travelled (VMT) change?

Data description

The data used are the high-resolution trajectory data collected by both on-board units and road side units. Given the trajectories, we can extract the origins and destinations (ODs) of the trips, which could be regarded as the first and the last locations of the trajectories. Due to the precision of the GPS devices and the "cold start" effect, the destination of the current trip might not match with the origin of the next trip of the same traveler. This is not a big issue in this context because it does not influence the general framework of the problem.

The left part of Figure 3.9 shows the distribution of the ODs of the trips in southeast Michigan collected from 2014 to 2018. The area with the highest travel demand of the Safety Pilot participants is the Ann Arbor-Ypsilanti area, highlighted by the red rectangle. Within all the trips, 79% of them has either origin or destination in the highlighted area, and within these trips, 81% of them have both origin and destination in the area. Therefore, our focus is on the highlighted area. By focusing on this area, the scale of the problem and the computational time required to solve the problem can be significantly reduced. The right part of Figure 3.9 shows the distribution of ODs in the selected area.

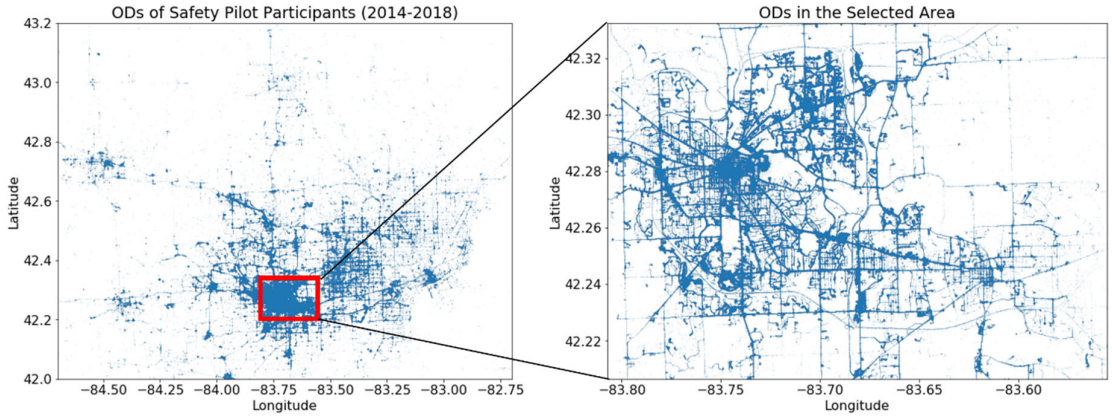


Figure 3.9 ODs of all the trips

Offline minimum fleet problem

Denote the set of trips by $\mathcal{T}$. For any trip $T_i \in \mathcal{T}$, it can be represented by a tuple $(t_i^p, t_i^d, l_i^p, l_i^d)$, where the elements represent pick-up time, drop-off time, pick-up location, and drop-off location, respectively.

Vehicle shareability network is a directed network whose nodes are the trips in $\mathcal{T}$ (Vazifeh et al., 2018). Two trips T_i and T_j are connected by an arc from T_i to T_j, if

$$t_i^d + t_{ij} \leq t_j^p,$$

where t_{ij} is the minimum connection time between l_i^d and l_j^p. In other words, arc (T_i, T_j) means that it is possible for a vehicle to pick up the passenger of trip T_j after dropping off the passenger of trip T_i.

A path cover of a network is a set of paths by which all the nodes are covered. A node-disjoint path cover is a path cover that any two paths in it share no common node. Solving the minimum fleet size problem is equivalent to solving the minimum node-disjoint path cover problem of the vehicle shareability network. The number of paths in the resultant path cover represents the minimum number of vehicles needed. Each path indicates the route the corresponding vehicle should follow. Figure 3.10 shows a vehicle shareability network for five trips. The arcs in red represent a minimum node-disjoint path cover of the network. Solving the minimum path cover problem is NP-hard in general. However, since the vehicle shareability network is acyclic, it can be transformed into a bipartite matching problem which can be solved efficiently.

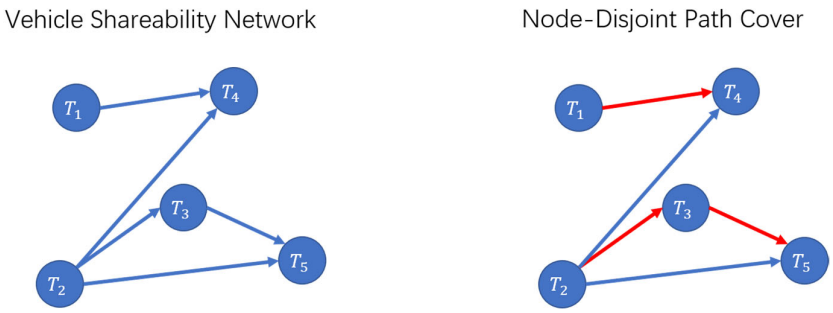


Figure 3.10 Vehicle shareability network and its node-disjoint path cover

Both the left column and the right column of the bipartite consist of the trips in $\mathcal{T}$. Based on the vehicle shareability network, the edges of the bipartite can be constructed correspondingly. Denote the cardinality of the max matching of the bipartite by m. Then, the minimum fleet size is $|\mathcal{T}| - m$. This is because in the left (right) column of the bipartite, there are $|\mathcal{T}| - m$ nodes with zero degree, representing the end (start) of the $|\mathcal{T}| - m$ routes. All the other nodes have a degree of two, indicating that they are in the middle of the routes. The maximal matching problem can be solved efficiently by Hopcroft-Karp algorithm (unweighted) or Kuhn-Munkres algorithm (weighted).

By setting the weights on edge (T_i, T_j') of the bipartite as $N - d_{ij}$, the solution can guarantee both minimum fleet size and minimum connection distance, where N is a sufficiently large number and d_{ij} denotes the distance between l_i^d and l_j^p,. Formally, the maximal matching problem can be expressed as

$$\begin{aligned} \text{maximize} \quad & \sum_{ij} (N - d_{ij}) x_{ij} \\ \text{subject to} \quad & \sum_i x_{ij} \leq 1, \forall i \\ & \sum_j x_{ij} \leq 1, \forall j \\ & x_{ij} \in \{0,1\}, \forall i, j \end{aligned},$$

where $x_{ij} = 1$ if the edge between trip T_i in the left column and trip T_j' in the right column is in the matching, 0 otherwise.

In Figure 3.11, the edges in red represent a max matching of the bipartite. Similarly, if the weights on the edges of the bipartite are set as $N - (t_j^p - t_i^d)$, then the solution of the max matching problem guarantees both minimum fleet size and minimum connection time.

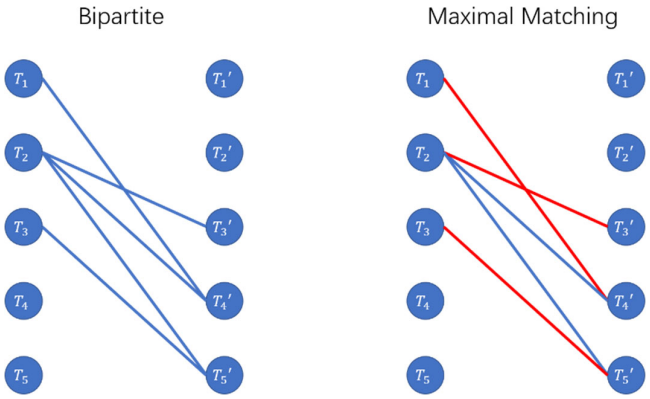


Figure 3.11 The equivalent bipartite and its maximal matching

Online minimum fleet problem

In the offline case, the ODs of all the trips are known in advance. In the online case, the information of the trips can only be known when a traveler sends a request to the fleet management system. Once there is a vehicle that can serve the request, the system will route the vehicle to pick up the traveler and deliver the passenger to the destination. The matching between vehicles and travelers can be formulated as an assignment problem and solved by bipartite matching algorithms (for instance, Kuhn-Munkres algorithm) efficiently.

When ridesharing is allowed, i.e., when a vehicle take two or more passengers at the same time, it is not a bipartite matching problem anymore. Denote the set of vehicles by V; denote the set of passengers by R; denote the set of trips by T. A trip is a subset of V, in which all the passengers' trips are shareable with each other. In other words, they can be potentially picked up by the same vehicle without violating any capacity or waiting time constraints.

Then an RTV graph can be built, as shown in Figure 3.12. In this graph, a passenger node is connected to a trip if it is a member of the trip; a vehicle and a trip is connected if the vehicle can take all the passengers in the trip (Alonso-mora et al., 2017).

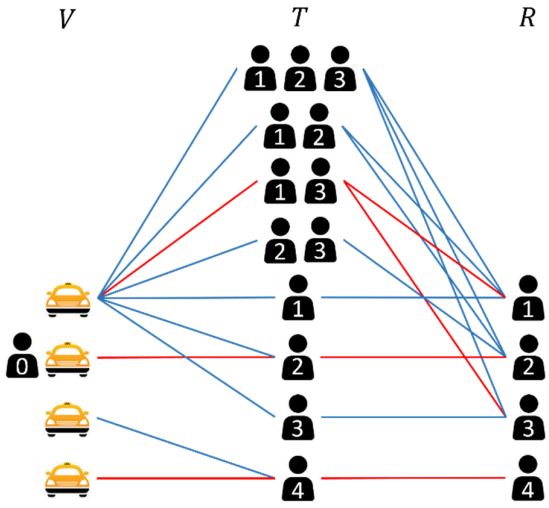


Figure 3.12 RTV graph in the online mode

For trip $t \in T$ and vehicle $v \in V$, set the weight on edge (t, v) as $w(t, v)$ which is usually related to distance or time between t and v. The goal is to maximize the sum of the weights, without violating any constraints. Formally, the problem can be formulated as the following mixed integer programming problem.

$$\begin{aligned} \text{minimize} \quad & \sum_{(t,v)\in A_{TV}} w(t,v)x_{tv} \\ \text{subject to} \quad & \sum_{t:(t,v)\in A_{TV}} x_{tv} \le 1, \forall v \in V \\ & \sum_{t:(t,r)\in A_{TR}} \sum_{v:(t,v)\in A_{TV}} x_{tv} \le 1, \forall r \in R \\ & x_{tv} \in \{0,1\}, \forall t \in T, v \in V \end{aligned},$$

where binary variable $x_{tv} = 1$ if trip $t \in T$ and vehicle $v \in V$ are matched, 0 otherwise. The first constraint ensures that each vehicle can serve at most one trip at the same time. The second constraint guarantees each passenger can only be assigned to at most one vehicle. For instance, the arcs in red in Figure 3.12 form a feasible solution of the problem.

Application to the Safety Pilot data

The methodology was applied to Safety Pilot Data collected during the year of 2015. All the ODs were extracted from the GPS trajectories. For the purpose of privacy protection, the real origins and destinations of the participants' trips were mapped to the closest intersections on the transportation network.

Figure 3.13 shows the comparison between the actual number of vehicles and the ideal minimum fleet size in each day of 2015. In the figure on the left-hand side, the blue curve represents the actual number of vehicles recorded in the Safety Pilot database. The orange curve represents the minimum fleet size required to complete all the trips if different families share vehicles. The peaks represent the weekdays and the valleys represent the weekends. The figure on the right-hand side implies that a fleet with 200 vehicles could almost satisfy the transportation demand every day. One can see that, ideally, vehicle sharing could significantly reduce the number of vehicles. The average number of vehicles could drop from 443.9 to 111.1.

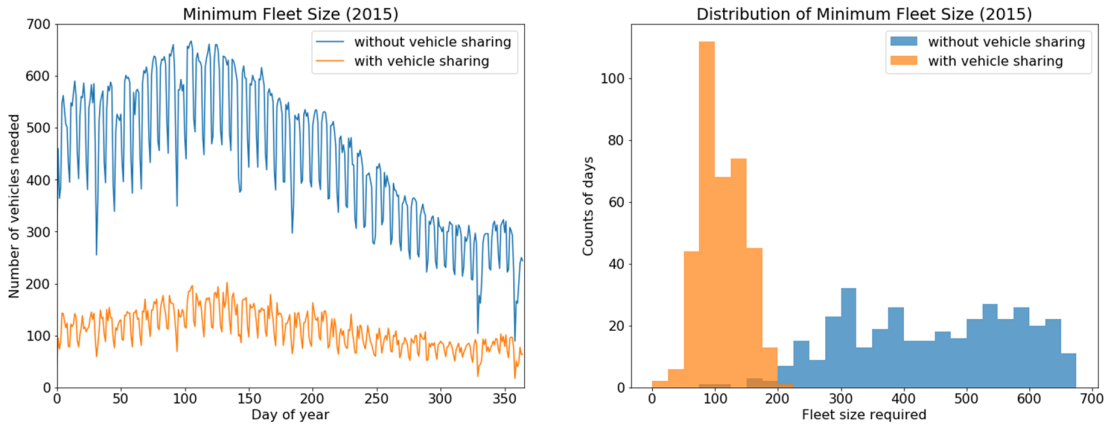


Figure 3.13 Actual number of vehicles and minimum fleet size

However, in terms of VMT, if different families share vehicles but do not share rides, the vehicles have to travel more because of the connections between different trips. Figure 3.14 shows the change of VMT. On average, VMT would increase by 25.6%.

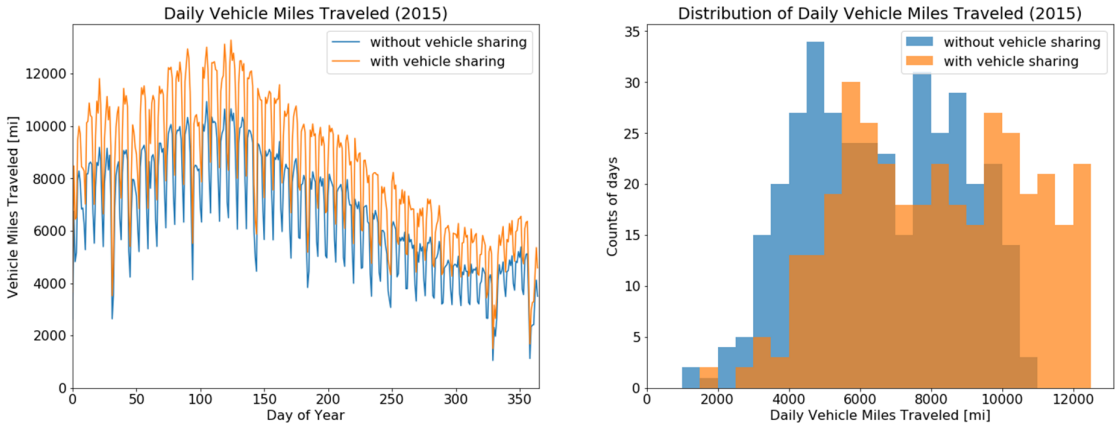


Figure 3.14 Actual VMT and VMT of minimum fleet size

Further analysis shows that to a large extent, the minimum fleet size is determined by the peak demand rate in each day. Figure 3.15 shows the correlation between the minimum fleet size and the maximum hourly demand in each day. The corresponding R^2 is 0.93, which implies a strong correlation.

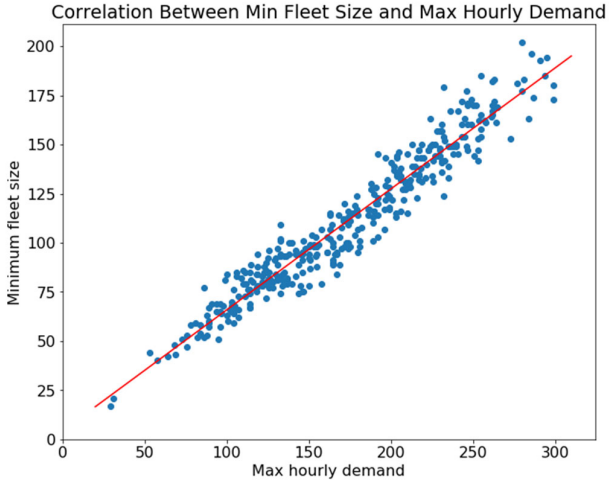


Figure 3.15 Correlation between min fleet size and max hourly demand

The extracted data are also used for the online case, where the ODs of all the trips are not known in advance. The left part of Figure 3.16 shows the number of served trips with different maximum waiting time, using the trip data on Feb 2, 2015 (Monday). Apparently, the more time the travelers are willing to wait, the more likely their requests will be served. It implies that even in the online mode, 120 vehicles are able to serve almost all the transportation demand, with a ten-minute max waiting time.

The right part of Figure 3.16 shows the results when not only are vehicles shared but also ridesharing is allowed. In general, the fleet size could be further reduced.

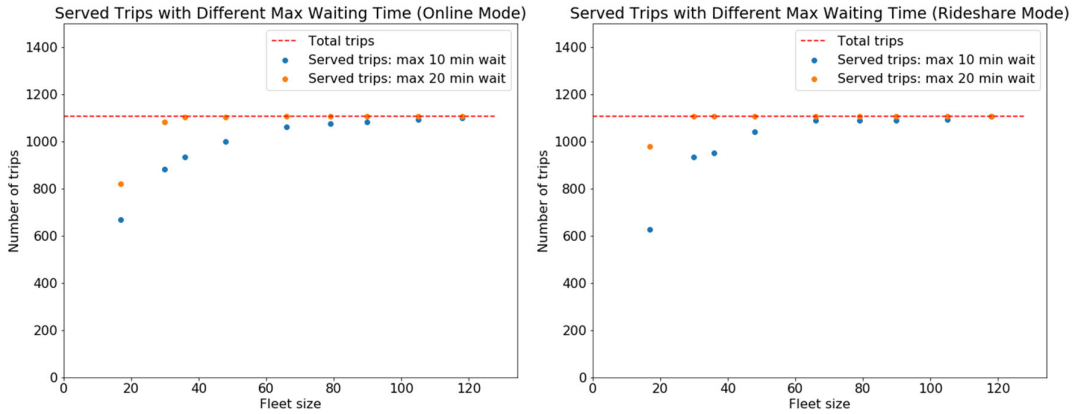


Figure 3.16 Trip service rate with different fleet size and waiting time (online mode and rideshare mode)

The above results are for the data collected from the Safety Pilot project, whose participants are sparsely distributed in southeastern Michigan. If the population density is higher, the effect can be even more significant. To further verify the methodology, it is applied to the taxi trip data in Manhattan, New York. Figure 3.17 shows the result for New York taxis. When ridesharing is allowed, the same fleet size can serve much more trips. In other words, the fleet size could be significantly reduced as well.

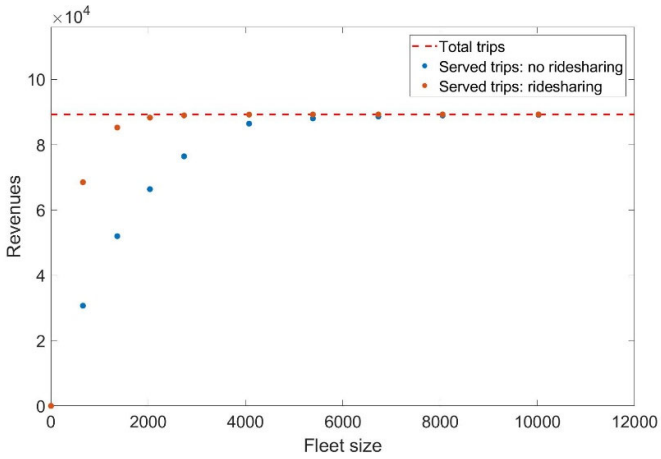


Figure 3.17 The effect of ridesharing on New York taxis

Summary

This section studies the potential change of the fleet size and VMT if different families share a fleet of vehicles. The offline problem is formulated as a node-disjoint path cover problem which is equivalent to solving for a weighted maximal matching on a bipartite. The solution of the bipartite matching problem can guarantee both minimum fleet size and minimum connection distance (or time). The online versions of the minimum fleet problem are also studied.

After applying the methodology to all the trips collected by the Safety Pilot project in 2015, the results show that if different families share a fleet of vehicles, the total number of vehicles needed could be significantly reduced (by 75.0%). However, the VMT would increase by 25.6%, due to the connections between different trips. Even in the online case, when the information of the trips is known in real time, 120 vehicles could serve almost all the trips if the travelers are willing to wait for ten minutes. Further analysis also shows that the minimum fleet size has a strong correlation with peak-hour demand rate.

References (for Task 3)

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E. and Rus, D., 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proceedings of the National Academy of Sciences, 114(3), pp.462-467.

Arnott, R., De Palma, A. and Lindsey, R., 1990. Economics of a bottleneck. Journal of urban Economics, 27(1), pp.111-130.

Baldassare, M., Ryan, S. and Katz, C., 1998. Suburban attitudes toward policies aimed at reducing solo driving. Transportation, 25(1), pp.99-117.

Caulfield, B., 2009. Estimating the environmental benefits of ride-sharing: A case study of Dublin. Transportation Research Part D: Transport and Environment, 14(7), pp.527-531.

Ester, M., Kriegel, H.P., Sander, J. and Xu, X., 1996, August. A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd (Vol. 96, No. 34, pp. 226-231).

Koppelman, F.S., Bhat, C.R. and Schofer, J.L., 1993. Market research evaluation of actions to reduce suburban traffic congestion: Commuter travel behavior and response to demand reduction actions. Transportation Research Part A: Policy and Practice, 27(5), pp.383-393.

Shaheen, S., Chan, N., Bansal, A. and Cohen, A., 2015. Shared mobility: A sustainability & technologies workshop: definitions, industry developments, and early understanding.

Teal, R.F., 1987. Carpooling: who, how and why. Transportation Research Part A: General, 21(3), pp.203-214.

Toole, J.L., Herrera-Yaqüe, C., Schneider, C.M. and González, M.C., 2015. Coupling human mobility and social ties. Journal of The Royal Society Interface, 12(105), p.20141128.

Vazifeh, M.M., Santi, P., Resta, G., Strogatz, S.H. and Ratti, C., 2018. Addressing the minimum fleet problem in on-demand urban mobility. Nature, 557(7706), p.534.

Vickrey, W.S., 1969. Congestion theory and transport investment. The American Economic Review, 59(2), pp.251-260.

Von Luxburg, U., 2007. A tutorial on spectral clustering. Statistics and computing, 17(4), pp.395-416.

Yang, H. and Huang, H.J., 1999. Carpooling and congestion pricing in a multilane highway with high-occupancy-vehicle lanes. Transportation Research Part A: Policy and Practice, 33(2), pp.139-155.

Zheng, Y., 2015. Trajectory data mining: an overview. ACM Transactions on Intelligent Systems and Technology (TIST), 6(3), p.29.

Task 4 System model development and Validation

The objective of Task 4 is to develop and validate a transportation system model of the Ann Arbor region, both for baseline analysis as well as CAV scenario analysis. Task 4 will incorporate data collected as a part of Task 1 and Task 2, as well as behavioral models developed in Task 3 into an integrated travel behavior and transportation system simulation model using the POLARIS software. The updated POLARIS model will then be used to simulate various connected vehicle technologies and quantify changes in energy consumption resulting from those scenarios.

The table shows the proposed subtasks and delivery times for the seven subtasks of Task 4, with a note regarding the expected completion time. The subtasks for Task 4 have not presented major unexpected obstacles.

Subtask	Subtask	Schedule	Schedule end date	Status

4.1	Implement baseline POLARIS model	M1-M12	9/16	Completed
4.2	Determine data needs for further model development	M1-M12	9/16	Completed
4.3	Query, collect and process data from the connected vehicle fleet	M13-M24	9/17	Completed
4.4	Implement traveler and CAV agent behavior rules	M25-26	11/17	Completed
4.5	Implement and calibrate the POLARIS-Autonomie model	M27-M30	3/18	Completed
4.6	Model validation with new data from field tests	M31-M32	5/18	Completed
4.7	Vehicle energy consumption quantification	M31-M36	8/18	Completed

Subtask 4.1. Implement baseline POLARIS model

The baseline POLARIS model for Ann Arbor has been fully implemented. Further development of the model will occur within task 4.4 and validation of the model will be completed under task 4.5

Subtask 4.2. Determine data needs for further model development

Completed

Subtask 4.3. Query, collect and process data from the connected vehicle fleet

The target is the development of a process querying, collecting and processing the data from vehicle test data in order to define the baseline for all our CAV & ML fuel consumption validation work. The procedure requires many database analysis, cleaning and restructuration.

4.3.1 Vehicles medialization

The University of Michigan provided so far 154 conventional vehicles, 52 HEVs, 13 PHEVs and 2 EVs which the manufacturing years are from 2011 and over (manufacturing before 2011 is not taken in consideration).

- 59 conventional vehicles are now modeled and validated (**less 5%** Fuel economy difference on the regulatory EPA cycles).
- 24 conventional vehicles and 30 HEVs are in validation process **(**Fuel economy difference on the regulatory EPA cycles between **5% and 10%**)

4.3.2 On-road test data

- Some inconsistencies found on data acquisitions and address to University of Michigan:
  - Potential errors in data acquisition
  - Some vehicle speed trace remains at zero
  - Some vehicles are not moving for the most part of the trip
  - Acceleration grade issues on cycles

- Validation of Autonomie's vehicle models on four real world driving cycles is shown in the table below

Cycle id	Distance [km]	Real world fuel cons [l/100km]	Autonomie fuel cons [l/100km]	Diff [%]	Real world fuel eco [mpg]	Autonomie fuel eco [mpg]
cycle_473_1	7.1	0.6	0.5	**16.2%**	28.2	31.3
cycle_473_2	13.4	1.0	1.0	**7.0%**	33.1	30.8
cycle_473_3	12.9	1.0	1.0	**0.0%**	30.0	29.9
cycle_473_13	14.0	1.0	1.0	**0.1%**	32.8	32.7

4.3.3 Next steps

- Continue models building and validation (on the regulatory EPA cycles & the Real World cycle)
- Work with University of Michigan on fixing the potential inconsistencies from on-road test data
- Run large scale simulation and validate the energy consumption

Subtask 4.4 Implement traveler and CAV agent behavior rules

4.4.1 Eco-Mobility on Demand Service with Ridesharing

Mesoscopic fuel consumption model

To evaluate the network-wide impact of Eco-Routing, an Eco-Routing algorithm is going to be implemented in POLARIS, as discussed in the previous quarterly update. In the application, the connected vehicles serve as probed vehicles and send motion information such as speed and acceleration to traffic management center, and the traffic management center broadcasts the vehicle motion information to the network. Host vehicle can use the broadcasted vehicle motion information and fuel consumption model to estimate the fuel-optimized route. One of the core functions is the fuel consumption model, which takes the vehicle motion information as input and output estimated fuel consumption for host vehicle on the links. The model need to have the ability for online calculation and accurate enough for route optimization.

The fuel consumption is obtained with Gaussian Mixture Regression[1]. With the objective of regression to maximize the conditional likelihood of output on the input variables as shown in equation (0.1), based on Bayesian law (0.2), maximizing total likelihood is equivalent since the input variable is independent of model parameter. With θ denote as the set of model parameter, X denote as the input, and Y denote as the output

$$\theta^* = \arg\max_\theta l(Y \mid X, \theta) = \arg\max_\theta \prod_i p(Y_i \mid X_i, \theta) \tag{0.1}$$

$$p(Y, X \mid \theta) = P(Y \mid X, \theta) P(X \mid \theta) \propto P(Y \mid X, \theta) \tag{0.2}$$

In this way, instead of minimizing the squared error of model output, a Gaussian Mixture Model (GMM) is obtained by maximizing the total likelihood of input and output. Since each component of the GMM is multivariate Gaussian distribution, the conditional distribution of output for each component also follows Gaussian (0.3)

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N(\mu,\Sigma), \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{21} \\ \Sigma_{12} & \Sigma_{22} \end{pmatrix}$$
$$\Rightarrow (X_1 \mid X_2 = x) \sim N\left(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x-\mu_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{22}\right) \quad (0.3)$$

For GMM where the probability density function is the weighted sum of individual Gaussian component, the expectation of output is

$$E(Y \mid X = x_0) = \sum \pi_i E_i(y \mid x_0)\phi_i(x_0) = \sum w_i(x_0)E_i(y \mid x_0) \quad (0.4)$$

where $w_i(x_0)$ is the posterior of component probability based on marginal distribution of the input variable, and $E_i(y \mid x_0)$ is the expectation from individual components. The input variables to the model are shown in Table 4.5. Model parameters for road sections with different speed limits are obtained individually.

Table 4.5 Fuel consumption Model Inputs

Model layer input	
Motion related features	**Link related features**
Average speed	Average grade
Speed change	Link Length

The model performance is evaluated against average speed model[2], power balance model[3] and Neural Network. The equations of the benchmarks are shown in Table 4.6.

Table 4.6 Benchmark Models

Average speed model	$f = t \times \exp(\beta_0 + \beta_1 v + \beta_2 v^2 + \beta_3 v^3 + \beta_4 v^4 + \beta_5 s)$
Power balance model	$f_k = t \times (\beta_0 v + \beta_1 va + \beta_2 sv + \beta_3 v^3)$
Neural Network	

where *f* is the expected fuel consumption, *v* is average speed, *a* is average acceleration, *s* is average grade, *t* is travel time on the road section, β_i are model parameters. The performance of the models is tested with expected fuel consumption estimated from microscopic fuel consumption model simulated data with real life speed and grade trajectories. The relative error distribution of the proposed model and the benchmarks are shown in Figure 4.18 and Table 4.7. The mean absolute percentage error of our proposed model is 10%. For R^2, our model and Neural network show close performance, and further fine tuning of the Neural network has potential to achieve better performance. However, compared with the enormous number of parameters to tune in the neural network, our model is non-parametric, in this way, there is no need for parameter tuning.

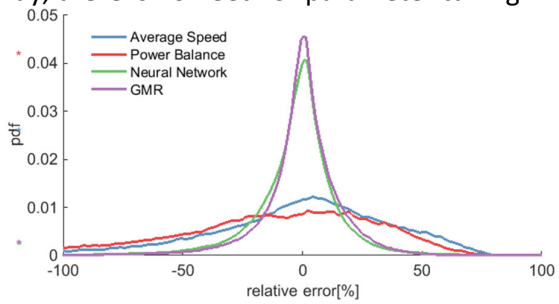


Figure 4.18 Model Performance Comparison

Table 4.7 Model Performance Comparison

	R^2	MAPE [%]
Average Speed	0.77	37.63
Power Balance	0.86	46.22
Neural Network	0.98	15.60
GMR	0.98	10.08

The fuel consumption estimation function is going to be implemented in POLARIS as cost estimation function in routing library and used together with current time-dependent A* routing policy to calculate eco route. Further investigation includes eco-routing with travel time constraint.

Constrained Eco Routing

To evaluate the benefit of eco-routing, a preliminary eco routing algorithm is implemented with Ann Arbor network in Matlab based on Dynamic Programming. Dynamic programming solves the optimization problem recursively based on Bellman principle

$$x_i^* = \arg\min_{x \in adj_{ont}(x_{i-1})} g(x_i) + f^*(x_{i-1}) \tag{0.5}$$

$$f^*(x_{i-1}) = \min_{x \in adj_{ont}(x_{i-2})} g(x_{i-1}) + f^*(x_{i-2}) \tag{0.6}$$

where x_i is the optimal next link location, x_{i-1} is the last link location, $f^*(x_{i-1})$ is the optimal value function at last link, $g(x_i)$ is the transition cost based on fuel consumption in traditional eco routing and weighted sum of travel time and fuel consumption in travel time constrained eco routing which is defined as

$$g(x_i) = (1 - w_t)c(x_i, x_{i-1}) + w_t t(x_i) \tag{0.7}$$

where $c(x_i, x_{i-1})$ is the estimated fuel consumption and $t(x_i)$ is the estimated travel time for current link. To address the travel time constraint, a soft constraint w_t is defined with respect to time limit t_c as shown in Figure 4.19.

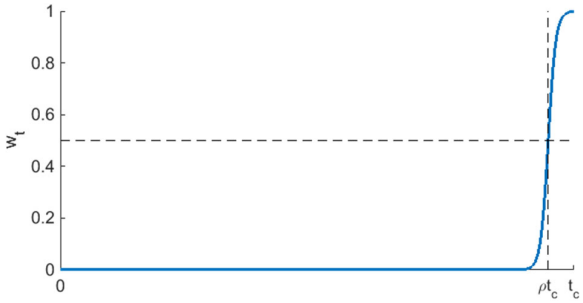


Figure 4.19 Soft constraint for travel time

The travel time constraint is defined as

$$t_c(x_i) = (1 + \varepsilon)t^*(x_i) \tag{0.8}$$

where ε is a relaxation constant and $t^*(x_i)$ is the travel time for shortest time path from origin to current link. Currently the fuel consumption and travel time are based on speed limit, and in the next step, real time traffic information is going to be taken into consideration. A case study comparing shortest path, fast path, eco-routing and constrained eco-routing are shown below in Figure 4.3.

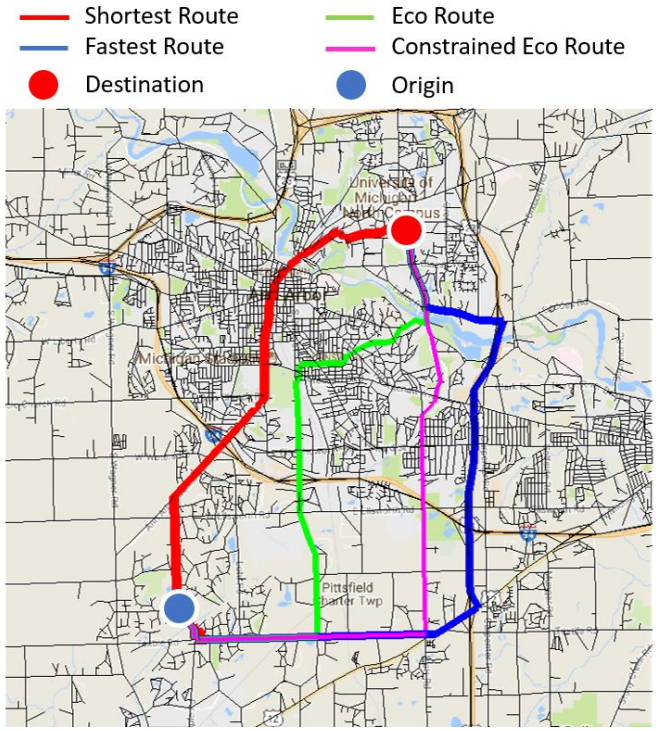


Figure 4.20 Routing results from different routing strategy

Table 4.8 Performance for different routing strategy in case study

	Fuel[kg]	**Time[s]**
Shortest	0.903	1050
Fastest	1.149	885
Eco Route	0.855	1081
Constrained	0.869	893

From the case study, it's shown that for the selected origin-destination pair, constrained routing has 24.4% reduction in fuel with 0.9% increase in time compared with fastest path. The routing results from one-to-all pairs are also evaluated as shown in Figure 4.21. The fuel consumption and travel time are normalized with fuel consumption of eco routing and travel time of fastest path respectively. The constrained eco routing can have a maximum 45% fuel consumption reduction with a maximum 4.4% increase in travel time compared with the fastest path.

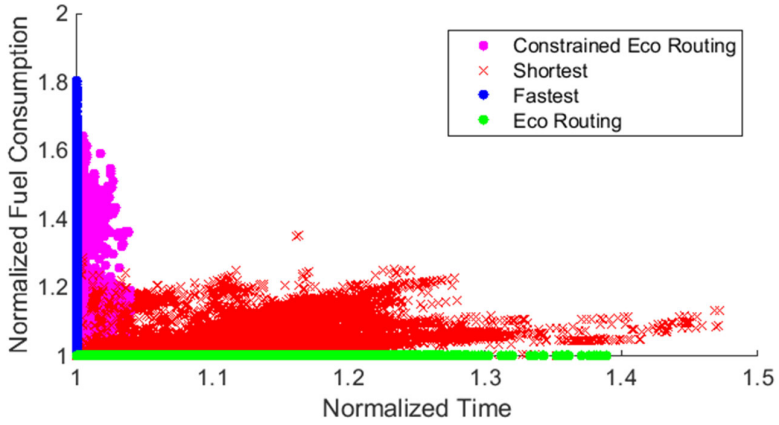
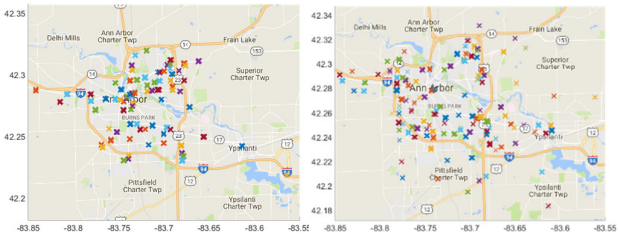


Figure 4.21 Normalized travel and fuel consumption from different routing strategy

In the preliminary implementation, the results are evaluated based on one-to-all route results, which means route from a single origin link to all other links. The next step is to evaluate the algorithm with the origin-destination pairs identified through Ann Arbor Connected Vehicle Pilot database and implement the routing algorithm in POLARIS to evaluate the benefit under real time traffic states.

Expected Benefit Estimation

To estimate the expected fuel consumption and travel time for different routing algorithms, we use travel origin-destination pairs from real-world driving data. We assume that the number of vehicles using proposed routing algorithm is limited, i.e., the vehicles cannot cause notable change to the travel speed of the links in the traffic network. The data to estimate travel demand is during May 2013 to October 2013, from 17:00 to 19:00 on weekdays. 25,001 trips were identified within the specified time. The origin and destination locations are identified through a density based cluster algorithm called OPTICS[4]. The advantage of this algorithm compared with other distance based clustering algorithms such as DBSCAN [5] is that it can cluster data with density change. This is critical in our analysis since the spatial densities of trip origin and destination locations can be affected by multiple factors such as parking lot size. We only include trips happening at least once per week. There are 3,031 frequently visited origin-destination pairs identified, and the identified starting and ending locations are shown in Figure 4.22.



(a) (b)

Figure 4.22 Trip locations identified with OPTICS: (a) Trip starting locations; (b) Trip ending locations

The studied Ann Arbor traffic network consists of 21,569 directed links with variate link types including local, minor, major, collector, ramp, and highway. The computation time to solve all-to-one routing result is around 13 s on a computer with Intel Core i7 and 16 G RAM. Considering requirement for the travel time of shortest-time routing, the computation time for constrained eco-routing is about 26 s. The routing cost are evaluated based on historical average speed during the studied hours. The uncovered links are imputed with their posted speed limits. Since they are never traveled by the sample vehicles over 6 months, we hypothesize these links are less traveled and the posted speed limit is a reasonable approximation for the free flow speed. To get the historical average speed, we use GMM to approximate average speed distribution of individual links and estimate the posterior of mixing coefficient based on speed during the sampled hours. The expectation of travel speed is estimated with the estimated posterior of the mixing coefficient.

To compare travel time and fuel consumption for different routing strategies, travel time and fuel consumption of different strategies are normalized with the travel time of fastest route and the fuel consumption of unconstrained eco-route respectively. The normalized costs are shown in Figure 4.23. The scatter plot is overlaid with expectation of cost estimated with the OD pair travel frequency. The error bars for each routing solution are 10% and 90% percentiles respectively. The expectation of travel time and fuel consumption are summarized in Table 4.9.

From the results, we can see that the shortest path consumed less fuel compared with the fastest path algorithm, while the travel time is increased significantly. Also, with a maximum of 6.48% increase in travel time, the constrained eco-routing solution has expected fuel saving of 5.16% and maximum saving of 51.8%, compared with the fastest-path solution. It's also noted that for the given OD pairs, 28% of the eco-routing solution are the same as the fastest-path solution, and 27% is the same as the shortest-path solution. For constrained eco-routing results, 55% is the same as the fastest-route solution and 27% is the same as the shortest-path solution. Besides that, 28% of shortest path and fastest-path are the same. The difference between eco-routing and constrained eco-routing is due to the travel time constraints.

Eco-Mobility-on-Demand Service with Ridesharing

As a start point, we reproduce the work in [6] by assuming the road network is static and solving all optimal routes considering travel time and fuel consumption offline. Travel time and fuel consumption of corresponding routing strategy is used for cost and constraints evaluation in assignment. Including dynamic road network information is done later. The trip assignment algorithm is based on a shareability graph. The graph is defined as undirected

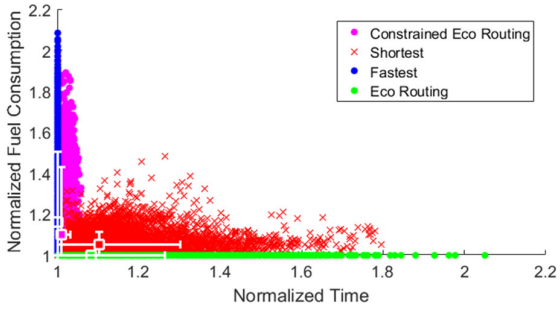


Figure 4.23 Normalized travel time and fuel consumption for different routing strategies

Table 4.9 Expected travel time and fuel consumption of different routing strategies

	Fuel consumption [kg]	Travel Time [s]
Shortest	0.4809	611.37
Fastest	0.5312	554.45
Eco-routing	0.4576	601.04
Constrained eco-routing	0.5038	559.49

graph with nodes defined as customers and vehicles. An edge exists between two customers if a vehicle can depart from the origin of one of the customers and fulfill the travel demands of both customers without violating travel time constraints. An edge exists between a vehicle and a customer if the demand can be served by the vehicle without violating travel time constraints. Then a necessary condition for a trip to be feasible is that the customers of the trip can form a clique with one vehicle present in the shareability network. A clique is a subgraph such that every node is connected to every other node within the same clique. It's noted that the cliques do not need to be maximum cliques in the shareability graph. The cliques in a graph can be found with Bron-Kerbosch algorithm [7] with worst case time complexity $O(dn3^{d/3})$ where n is the number of nodes and d is degeneracy of the graph, which is a measure of sparseness. In this way, instead of evaluating cost of trips for every possible combination of customers and vehicles, one can solve single-vehicle-multiple-customer problems for every clique.

Trip scheduling for each clique is a traveling salesman problem with pickup and delivery. The problem can be solved with multiple algorithms. If the number of customers is small, (e.g., less than 5), the exact solution can be found by Dynamic Programming in less than 1 sec on a standard desktop computer. Heuristic based algorithms such as T-share [8] can be used to find the solution if the problem size is large.

After all feasible trips were found through solving the scheduling problem for all cliques, the optimal trip assignment problem can be formulated and solved through Integer Linear Programming (ILP). In this Section, we briefly summarize the formulation from [6].

The cost for each customer consists of wait time and delay time. Wait time is defined as time between the customer travel request and time of pickup. Delay time is defined as the difference between planned travel time and the shortest travel time after pickup, which is from the fastest path solution from origin to destination. The cost of a trip is defined as wait time plus delay time for all customers if fastest routing strategy is used, and fuel consumption if eco-routing strategy is selected, denoted as c_t^i for trip i. The states of the system are δ_t which is the indicator variable for trip/clique and δ_c which is the indicator variable for a customer. If at an assignment instant, there are m feasible trips from TSP step and n customers, then $\delta_t = \{\delta_t^i \in \{0,1\}, i \in \mathbb{N}, 1 \leq i \leq m\}$ and $\delta_c = \{\delta_c^i \in \{0,1\}, i \in \mathbb{N}, 1 \leq i \leq n\}$. δ_t^i is 1 if trip i is selected and δ_c^i is 1 if customer i is assigned. The objective function is

$$\sum_{i=1}^{m} c_t^i \delta_t^i + \sum_{i=1}^{n} D\left(1 - \delta_c^i\right), \tag{0.9}$$

where D is the penalty for unserved customers. The constraint for vehicle is that each vehicle can only serve one trip

$$\sum_{i=1}^{m} a_j^i \delta_t^i \leq 1, \forall j, \tag{0.10}$$

where a_j^i is the indicator variable for vehicle j and trip i, $a_j^i = 1$ if vehicle j can serve trip i. The constraint for customer is that a customer is either assigned or ignored

$$\sum_{i=1}^{m} b_j^i \delta_t^i + \left(1 - \delta_c^j\right) = 1, \forall j, \tag{0.11}$$

where b_j^i is the indicator variable for customer j and trip i, $b_j^i = 1$ if customer j can be served by trip i. With linear constraints and the objective function, the trip assignment problem is an integer linear programming. For online optimization, we follow [6] to keep a pool of customers and a customer is removed from the pool if it's picked up by vehicle or cannot be served within the time constraint. If a customer is ignored, a vehicle from the idling fleet is assigned to serve the vehicle with minimum wait time as the objective.

We randomly selected 4% of the trips generated during the studied time as demand for the shared mobility fleet. The trip generation rate is 35~40 new trips every 30 sec and we follow the re-optimization strategy every 30 sec from [6]. The simulation period for our study is 30 min. We fix the fleet size at 900 and the vehicle capacity is 4. The wait time constraint is 3 min, and the delay time constraint is 3 min. The benchmark algorithm is the non-sharing case, where everyone drives their own cars. For ridesharing control, 3 different strategies are selected and the routing strategies in corresponding phase are summarized in Table 4.10.

Table 4.10 Routing Strategy of Different MOD Control Strategies

MoD Control Strategy	**Assignment**	**Rebalance**
Fastest routing	Time	Time
Eco-routing	Fuel	Fuel
Hybrid routing	Fuel	Time
No sharing (baseline)	Time	-

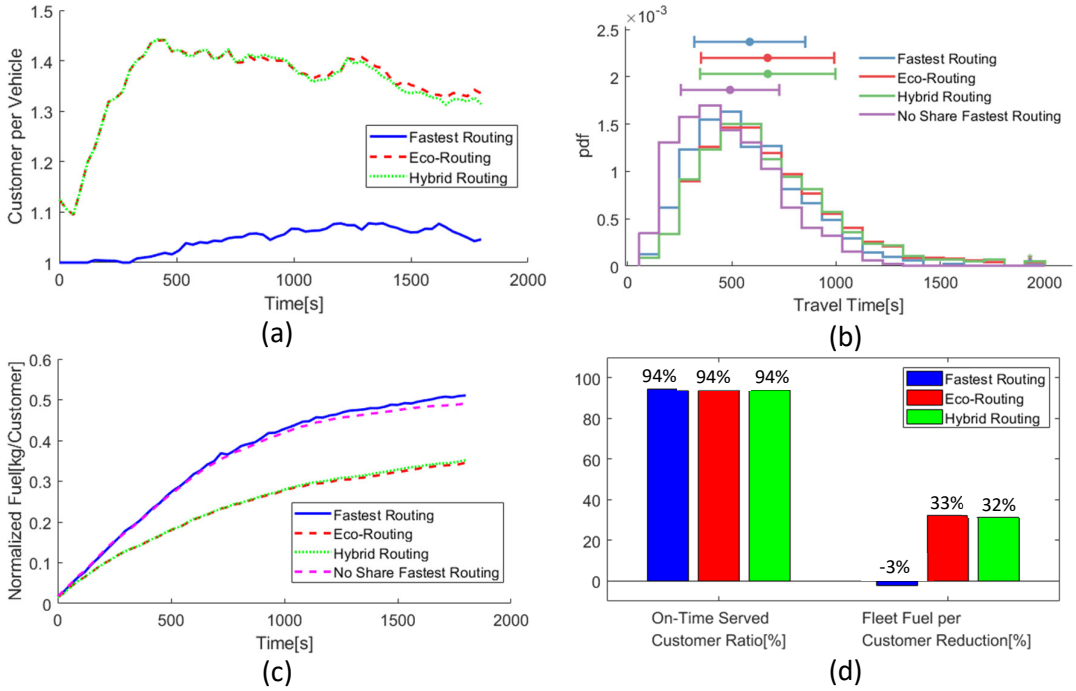


Figure 4.24 Simulation Results (a) Average customer number per vehicle; (b) Total travel time (c) Fleet fuel consumption per customer; (d) On-time served customer ratio and fleet fuel reduction compared with non-sharing baseline

During the simulated period, all customers are served. As shown Figure 4.24(a), our proposed algorithm results in 1.36 customers per vehicle, indicating more efficient usage of the fleet. Besides that, as shown in Figure 4.24 (c), with normalized fuel defined by fleet total fuel consumption normalized with number of served customers, the MOD service reduces the operation cost for the service provider. The total travel time increased as shown in **Error! Reference source not found.**(b), however, it should be noted that all algorithms can serve more than 94% of customers within the time constraints. With travel time as cost, the total fuel consumption is increased by 3% due to the increased vehicle travel mileage in rebalancing and picking-up. If fuel consumption cost is considered by the service provider, the total fuel consumption can be reduced by more than 30%.

References (For Task 4.4.1)

[1] H. G. Sung, “Gaussian Mixture Regression and Classification,” Rice University, 2004.

[2] K. Boriboonsomsin, M. Barth, S. Member, W. Zhu, and A. Vu, “ECO-Routing Navigation System based on Multi- Source Historical and Real-Time Traffic Information,” *Network*, vol. 13, no. 4, pp. 1694–1704, 2012.

[3] J. Kwon, A. Rousseau, and P. Sharer, “Analyzing the uncertainty in the fuel economy prediction for the EPA MOVES binning methodology,” *SAE Int.*, 2007.

[4] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, “Optics: Ordering points to identify the clustering structure,” *ACM Sigmod Rec.*, pp. 49–60, 1999.

[5] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” *Proc. 2nd Int. Conf. Knowl. Discov. Data Min.*, pp. 226–231, 1996.

[6] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, “On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment,” *Proc. Natl. Acad. Sci.*, vol.

114, no. 3, pp. 462–467, Jan. 2017.
[7] C. Bron and J. Kerbosch, "Algorithm 457: finding all cliques of an undirected graph," *Commun. ACM*, vol. 16, no. 9, pp. 575–577, 1973.
[8] S. Ma, Y. Zheng, and O. Wolfson, "Real-Time City-Scale Taxi Ridesharing," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 7, pp. 1782–1795, 2015.

4.4.2 Eco-approach and Departure (EAD)

Motivation for Eco-approach and Departure

Traffic congestion in urban driving environments costs Americans an extra 6.9 billion hours of time and $160 billion worth of fuel annually [1]. Intersections where drivers rapidly decelerate to obey the traffic signals are among the most common area that result in increased fuel use and travel time. Connected Vehicle (CV) technologies offer an opportunity to resolve this problem by connecting vehicles and traffic infrastructure through wireless communication methods such as Dedicated Short Range Communications (DSRC). In this setting, vehicles can obtain information of other vehicles, and of traffic infrastructures, such as signal phasing and timing (SPaT) information of adjacent traffic lights. An eco-driving concept at signalized intersections, commonly called Eco-approach and Departure (EAD), has been proposed to solve the aforementioned problem by utilizing SPaT. The goal of EAD is to determine energy efficient trajectories of a vehicle as it passes through signalized intersections.

Motivation for the case study

Recent research which utilized SPaT on eco-driving at urban-driving environments have verified its potential to reduce fuel consumption, however, the results have been limited to simulations and controlled lab tests. The goal of the case study is a two-fold. First goal is to develop a realistic algorithm for eco-driving at urban-driving environments. The second goal is to verify and validate the benefits of the proposed algorithm in realistic settings.

In order to achieve the first objective, a realistic EAD algorithm should consider not only the host vehicle, but also surrounding environments such as the vehicles in front of the host vehicle, queues at the signalized intersections, vehicles in the next lanes, and stochasticity of traffic lights. Accordingly, we propose a novel EAD algorithm which takes the vehicles in front of the host vehicle, the queue length, and stochasticity of traffic lights into account.

For the second objective, we verify the benefits of the proposed method by comparing them to naturalistic human driving records at signalized intersections, instead of comparing the results to simulation models or controlled lab tests. Hundreds of human-driven trips made on Plymouth Road and Fuller Road, Ann Arbor, MI were extracted from the Safety Pilot Model Deployment (SPMD) [2] database, which contains naturalistic driving records of roughly 3,000 drivers in Ann Arbor, MI. The uniqueness of each driving records and SPaT makes each trip a unique problem. For each trip undertaken by a human driver, the proposed EAD method is used to produce a solution given the same initial and final conditions.

The EAD algorithm for free-flow traffic

We first propose a Dynamic Programming-based algorithm for free-flow traffic. Then, we compare its solutions to real-world results obtained for 609 human-driven trips in Ann Arbor, MI.

The problem is formulated as a non-linear optimization problem in which the objective is to find a fuel-efficient speed profile. It's system dynamics, cost function, and equality and inequality constraints of the problems are elaborated below. It is assumed that the vehicle is operating in free-flow traffic and equipped with a continuously variable transmission and an internal combustion engine whose fuel consumption is represented by a static brake specific fuel consumption map.

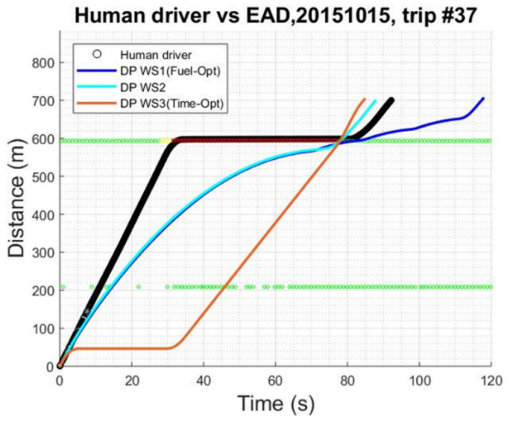
- **State** $x(t) \coloneqq [d(t),\ v(t)]'$
- **Input** $u(t) := a(t)$
- **System Dynamics** $Ma(k) = F(k) - Mgf - 0.5\rho C_d A v(k)^2$ (zero-grade & zero-wind speed)

- **Constraints** $a_{brake,max} \le u(t) \le a_{accel,\ max}$, $0 \le v(t) \le v_{max}$,
  $d(t_i) = d_i,\ t_i \in t_{green,\ i},\ i = 1,2,\dots n,$ $t_f \le t_{max,\ simulation}$
- **Initial & Final Conditions** $x(t_0) = [d_0, v_0],\ d(t_f) \ge d_f,\ v(t_f) \in [v_{nom} - \epsilon\ ,\ v_{nom} + \epsilon]$
- **Cost Function** $f(t)$ **:** Combination of fuel consumption, travel time, and riding comfort

$$f(t) \coloneqq w_{TT}\Phi(t_f) + \int_0^{t_f}(w_{FC}L_{FC}(t) + w_{RC}L_{RC}(t))$$
$$\Phi(t_f) \coloneqq t_f,\ L_{RC}(k) \coloneqq u(t)^2,\ L_{FC}(k) = \mathrm{FC}(\tau_{eng}(t),\ w_{eng}(t))$$

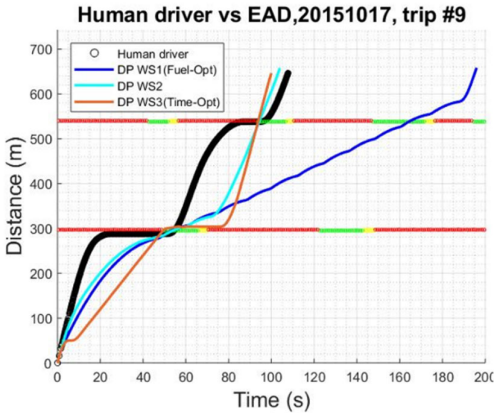
Results for free-flow traffic

For each human-driven trip, the proposed method uses Dynamic Programming to determine globally optimal trajectories of three different eco-driving policies: fuel-optimal policy (DP WS1), time-optimal policy (DP WS3), and nominal eco-driving policy (DP WS2). Given the same initial and the final conditions as those of a human driving record, comparisons are made across the driving records and the eco-driving policies to demonstrate the real-world benefits. Among 609 trips, two representative cases were selected and depicted in Figure 4.1, and Figure 4.2.



	Human	WS1	WS2	WS3
Fuel	35.4g	7.4g	8.2g	80.1g
Time	92s	118s	88s	85s

Figure 4.1. A representative trip on Fuller Rd, Ann Arbor [3]



	Human	WS1	WS2	WS3
Fuel	32.8g	12.4g	16.5g	60.2g
Time	108s	196s	104s	100s

Figure 4.2. A representative trip on Plymouth Rd, Ann Arbor [3]

As shown in the above figures, the amounts of fuel saving that the fuel-optimal policy and the nominal policy have are significant. Two explanations for how the proposed method saves fuel are as follows: First, EAD utilizes SPaT in planning the trajectories to minimize unnecessary acceleration and deceleration. Second, EAD searches solutions over the BSFC map of a vehicle and runs the engine of the vehicle at highly efficient engine operating points. The Figure 4.3 compares the engine operation points of a human driver and those of the nominal EAD policy for the trip depicted in Figure 4.2.

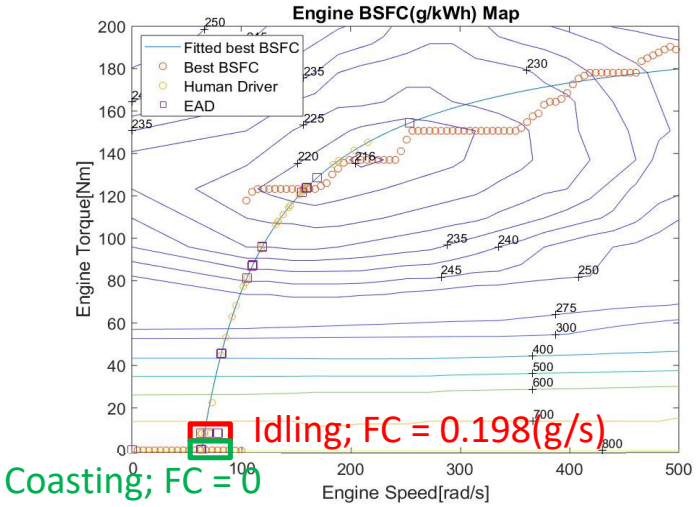


Figure 4.3. The engine operation points on BSFC map of the trip Figure 4.2. The engine operation points of the human driver are colored yellow, and those of the nominal EAD solution are depicted in purple.

It is shown that the human driver runs the engine at less fuel-efficient operating points, as he or she hit the idling engine operation point far longer than the EAD solution. The nominal EAD policy instead either coasts consuming its kinetic energy (thus consuming zero fuel) or hit one of the highly efficient operation points in the fuel map.

Total 287 and 322 human-driven trips were made on Fuller and Plymouth Road are compared with the three EAD policies. The fleet statistics on the nominal EAD policy are described in Figure 4.4 and 4.5, showing the potential fuel savings of 40-50% while matching human travel time.

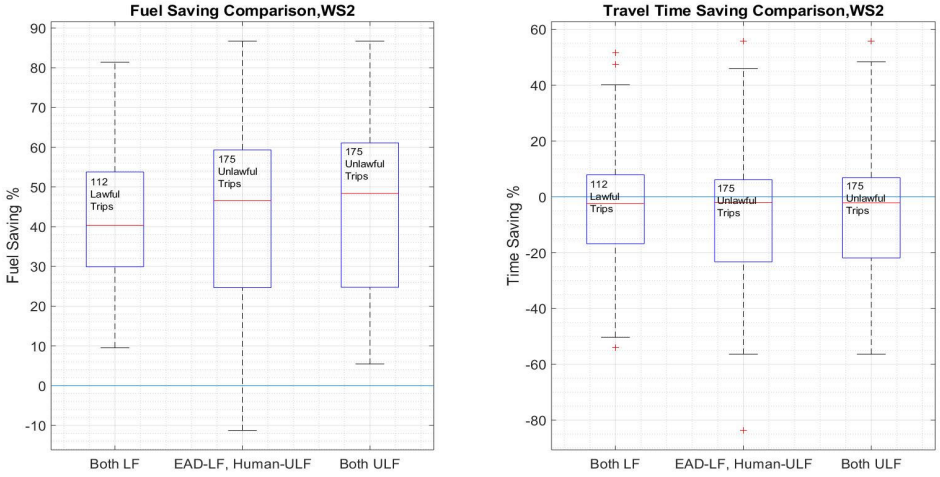


Figure 4.4. The nominal EAD policy solutions are compared with 287 human-driven trips undertaken on Fuller Road. For the Lawful case, EAD saves roughly 40% of fuel on average, while sacrificing travel time only 2.5%. For the cases of Unlawful human trips & Lawful EAD, the fuel saving benefit becomes smaller. For the Unlawful human and EAD case, the fuel saving greatly increases to achieve 48% [3]

In Figure 4.4 and 4.5, the first and the fourth box plots represent comparisons between fuel consumption and travel times of 'Lawful' EAD and 'Lawful' human drivers, i.e. those drivers who abided by the operating speed limit of 40 mph. The second and the fifth box plots represent those of 'Lawful' EAD and 'Unlawful' human drivers who exceeded 40 mph at some point over the trip. In order to draw a consistent comparison, the third and the sixth plots, represent the results when EAD is allowed to exceed the limit and its maximum speed is set to be the same as that of the human driving record.

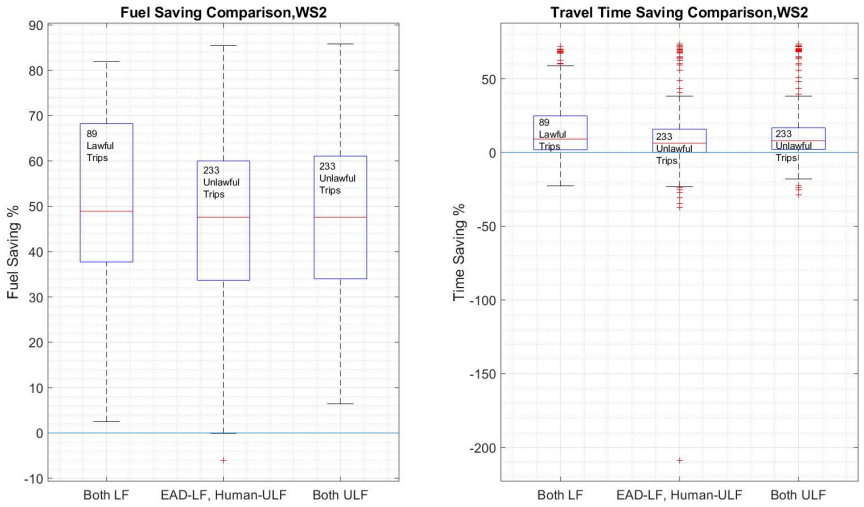


Figure 4.5. Comparisons between the nominal EAD policy solutions and 322 human-driven trips undertaken on Plymouth Road. Because of more frequent changes in traffic signals and longer red lights, the fuel and travel time saving potential is greater than that of Fuller Road. For the Lawful case, EAD not only saves fuel by 49% on average but also travel time by 10% [3]

The realistic EAD algorithm

The proposed eco-driving method for free-flow traffic showed potential fuel savings of 40-50% while matching human travel time when compared to roughly 600 naturalistic Human driving data. While the results can serve as an upper bound of fuel saving potential of the eco-driving in the vicinity of signalized intersections, they are limited to free-flow scenarios where the EAD vehicle is unconstrained by other vehicles. In many urban driving scenarios, the motion of a vehicle is constrained by other vehicles. The most important factor which constrains the motion of an EAD vehicle is the presence of a vehicle in front. The figure on the left illustrates how the presence of a front vehicle can affect the operation of the host vehicle, and how EAD path is changed accordingly.

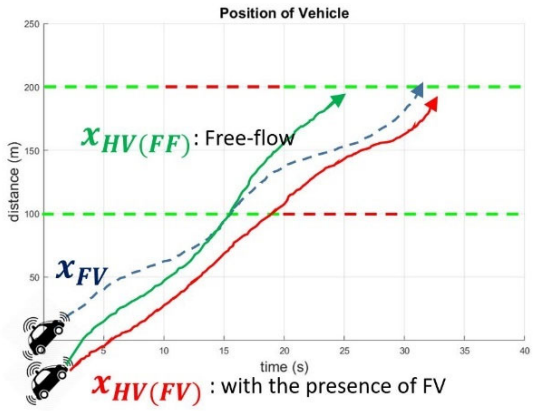


Figure 4.6. Illustration of different EAD solutions with/without a front vehicle [4]

Here, we assume that the front vehicle is a Human-driven vehicle. In order to find the optimal trajectory of the host EAD vehicle, the path of the Human-driven front vehicle should be obtained to calculate a safe set $S(t)$ of the host vehicle. A safe set describes combinations of the state, and the input which does not jeopardize the safety of EAD vehicle. The front vehicle EAD problem is formulated similar to the free-flow problem. An additional constraint on the state and the input is added to ensure the safety.

- **State** $x(t) \coloneqq [d(t),\ v(t)]'$
- **Input** $u(t) := a(t)$
- **System Dynamics** $Ma(k) = F(k) - Mgf - 0.5\rho C_d A v(k)^2$ (zero-grade & zero-wind speed)
- **Constraints** $a_{brake,\,max} \le u(t) \le a_{accel,\,max}$, $0 \le v(t) \le v_{max}$, $d(t_i) = d_i,\ t_i \in t_{green,\,i},\ i = 1,2,\dots n,$ $t_f \le t_{max,\,simulation}$
- **Safety Constraints** $[d(t),\ v(t),\ a(t)] \in S(t)$ $\forall t$
- **Initial & Final Conditions** $x(t_0) = [d_0, v_0],\ d(t_f) \ge d_f, v(t_f) \in [v_{nom} - \epsilon\ ,\ v_{nom} + \epsilon]$

- **Cost Function** (t) **:** Combination of fuel consumption, travel time, riding comfort, and value of a safety metric.

$$f(t) \coloneqq w_{TT}\Phi(t_f) + \int_0^{t_f}(w_{FC}L_{FC}(t) + w_{RC}L_{RC}(t) + w_S L_S(t))$$
$$\Phi(t_f) \coloneqq t_f,\ L_{RC}(k) \coloneqq u(t)^2,\ L_{FC}(k) = \mathrm{FC}(\tau_{eng}(t),\ w_{eng}(t))\ ,\ L_s(k) = \mathrm{Y}(d(t),\ v(t))$$

In addition to the hard constraint on safety, a safety metric is introduced in the cost function. Time-to-collision (TTC) and Time-headway (TH) are popular safety measures of the motion of vehicles. However, neither is appropriate as the safety constraint for EAD problem. First, large TTC does not necessarily mean 'safer'. In case of $v_{FV} \approx v_{HV}$, TTC can be big even if $x_{FV} \approx x_{HV}$, which indicates there was a crash. Second, TH can robustly guarantee the safety, but it may discourage the performance of EAD since TH calculate safety distance uses only v_{HV}. Therefore, a new safety measure for the cost function based on both v_{HV} and v_{FV} is proposed [4] and used to calculate the safety set. The equations for the three safety measures are described below. R_{des} is the desired range $(d_{FV} - d_{HV})$, A is stand-still separation distance, T_h is time-headway, B and C are real numbers.

- **Time-to-collision** $TTC \coloneqq \frac{x_{FV} - x_{HV} - l_{HV}}{v_{HV} - v_{FV}} \quad \forall\ v_{HV} > v_{FV}$
- **Time-headway** $R_{des} \coloneqq A + T_h v_{HV}$
- **Proposed safety measure** $R_{des} \coloneqq A + B(v_{HV}) + Cmin(0, (v_{HV} - v_{FV}))$

The prediction models

Predicting those of a Human-driven vehicle ahead of time is non-trivial. Since every human driver exhibits different driving patterns, and reacts differently to traffic signals, it is hard to write down an equation which describes the policy of Human-driven vehicles at signalized intersections. Thus, the Human policy model is driven by data and it is unique to each signalized intersection. In this sense, a Human acceleration policy model $a = f(X)$ is proposed [4].

Methodology	Test Set		
	MAE (m/s^2)	$\sigma^*(m/s^2)$	R^2
SVM, Gaussian	0.44	0.55	0.31
Boosted Trees	0.37	0.46	0.45
Random Forest	0.36	0.45	0.51
Artificial Neural-net	**0.36**	**0.45**	**0.50**

Table 4.1. List of prediction models and their performance. Assuming that acceleration of human driver follows a normal distribution, σ^* can be obtained from mean absolute error, $\sigma = \sqrt{\frac{\pi}{2}}\mathrm{E}[|\mathrm{X}|]$ [5].

A number of regression and classification models were obtained through supervised learning on the SPMD data. The models that we studied include linear model, Support Vector Machine with linear, quadratic, cubic, and gaussian kernel, decision trees, random forest, and neural-network. Then their performances were evaluated on a test set, as described in Table 4.1.

The standard deviations provide interpretability of the numbers. Assuming $a_x \sim N(\overline{a_x}, \sigma)$, the standard deviation of 0.45 m/s^2 corresponds to that of 1.42 m/s in the speed 10 seconds into the future with the assumption of constant acceleration for $t = [n, n+1)$ where n is an integer. We also assume that $a_x(t_1)$ is independent of $a_x(t_2)$ for all combinations of t_1 and t_2. Since the sum of two independent normally distributed random variables is normal, with its mean being the sum of the two means, and its variance being the sum of the two variances, the predicted speed of a vehicle at $t = k$ can be obtained as below.

$$v(k|t) = a_{x(0|t)} + a_{x(1|t)} + \cdots + a_{x(k-1|t)}$$

$$v(10|t) = a_{x(0|t)} + \cdots + a_{x(9|t)} = N\left(\overline{a_{x(0|t)}}, {\sigma_{x(0|t)}}^2\right) + \cdots + N\left(\overline{a_{x(9|t)}}, {\sigma_{x(9|t)}}^2\right)$$

$$= N\left(\overline{a_{x(0|t)}} + \cdots + \overline{a_{x(9|t)}}, {\sigma_{x(0|t)}}^2 + \cdots + {\sigma_{x(9|t)}}^2\right) = N\left(\overline{a_{x(0|t)}} + \cdots + \overline{a_{x(9|t)}}, \mathbf{1.42}^2\right)$$

While a number of assumptions are required to obtain the above results, it provides an useful insight on what the MAEs reveals. For example, an acceleration model with a MAE of 0.36 m/s^2 has a standard deviation of 0.45 m/s^2. The predicted speed into the future 10 seconds will then be normally distributed with the standard deviation of 1.42 m/s. Given that the mean of predicted accelerations is close enough to the true value, such predictions provide a mean of probabilistic interpretations.

The obtained models then used to predict trajectories of human vehicles at signalized intersections. Prediction examples are given in the figure below. Note that all $d(t)$, $v(t), a(t)$ described in the figure were obtained through 1 iteration of the prediction process, $d(t|t_0 = 0), v(t|t_0 = 0), a(t|t_0 = 0)$, $\forall t \in [0,40]$ for the top example and $\forall t \in [0,60]$ for the bottom example. It is worth mentioning that iterative predictions significantly increase accuracy of prediction.

Among the models we studied, the neural-network is the most complex with high nonlinearity, and the linear models are the least complex model. Based on the performance study which indicated that the neural-net and the random forest perform the best, we selected neural-network prediction model as our model.

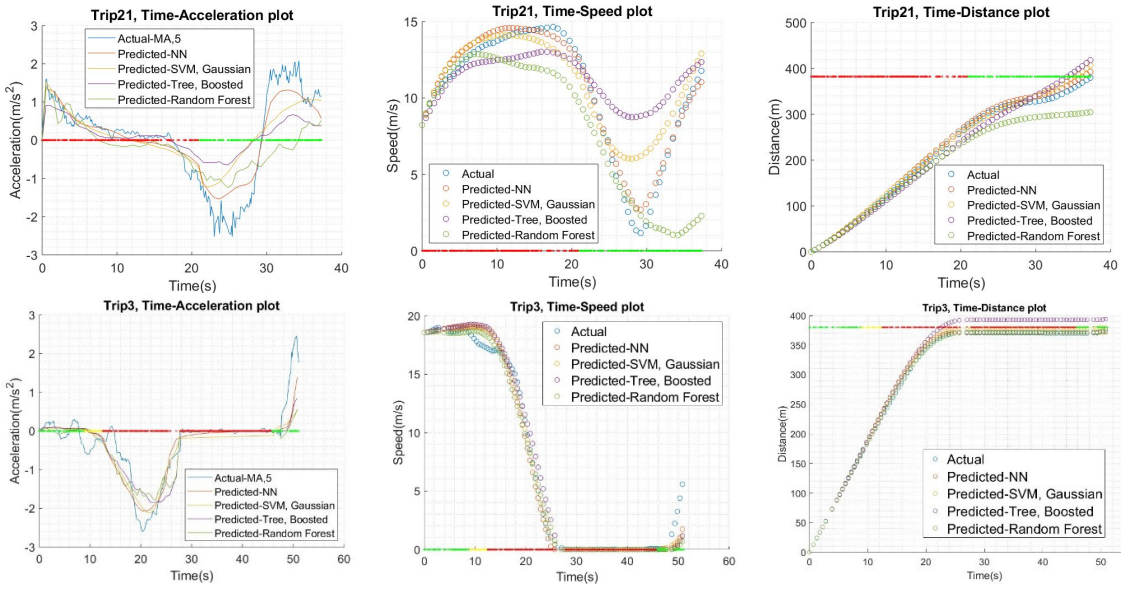
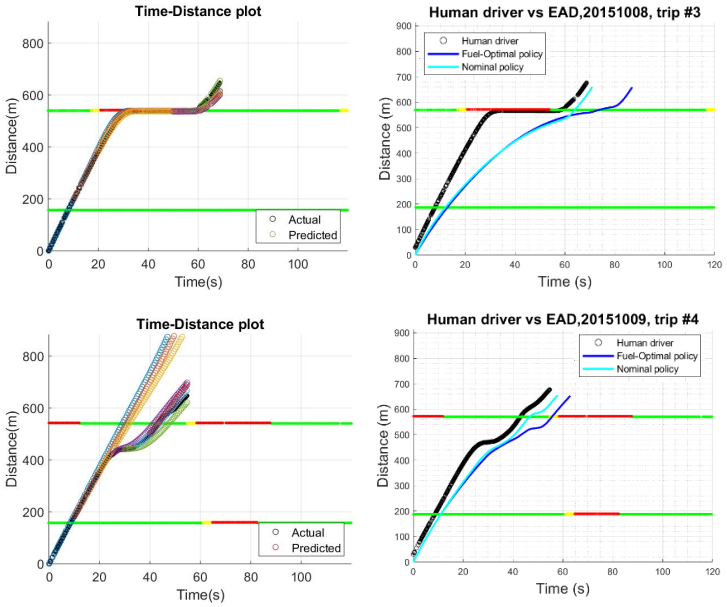


Figure 4.7. The left plots depict recorded accelerations of human-driven vehicle and predicted acceleration using 4 different prediction models listed in Table 4.1. The plots in the middle describe the predicted speed trajectories calculated based on the predicted acceleration. The right plots calculate the predicted vehicle trajectories based on the predicted speed profile. It is shown that the trend of the true accelerations and that of predicted accelerations match well, allowing the model to produce accurate predictions [4].

Result for non-free-flow cases

The prediction model then was incorporated into the problem and used to formulate the dynamics of the model and the corresponding optimization problem. Then the optimization problem is solved using approximate dynamic programming and model predictive control. An example result is described in Figure 4.8.



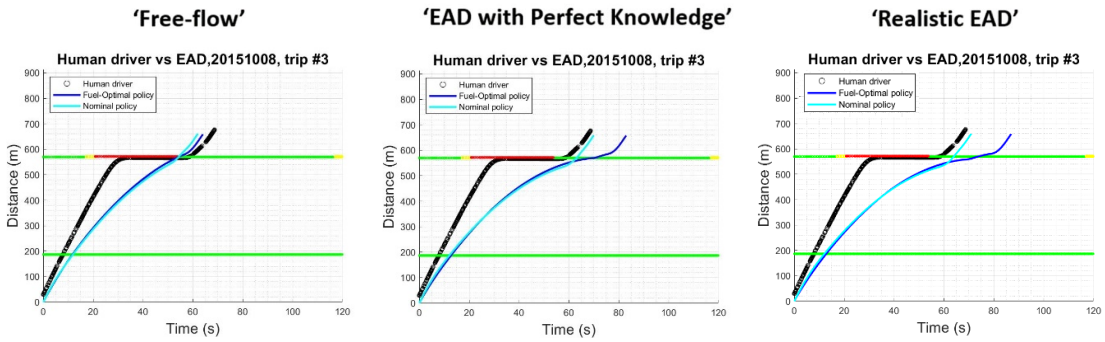
	Human	Fuel Optimal	Nominal	Re-plan
Fuel	25.8g	10.8g	11.6g	2sec
Time	68.6s	87s	71s	2sec

	Human	Fuel Optimal	Nominal	Re-plan
Fuel	25.9g	8.6g	16.3g	1sec
Time	54.6s	62s	58s	1sec

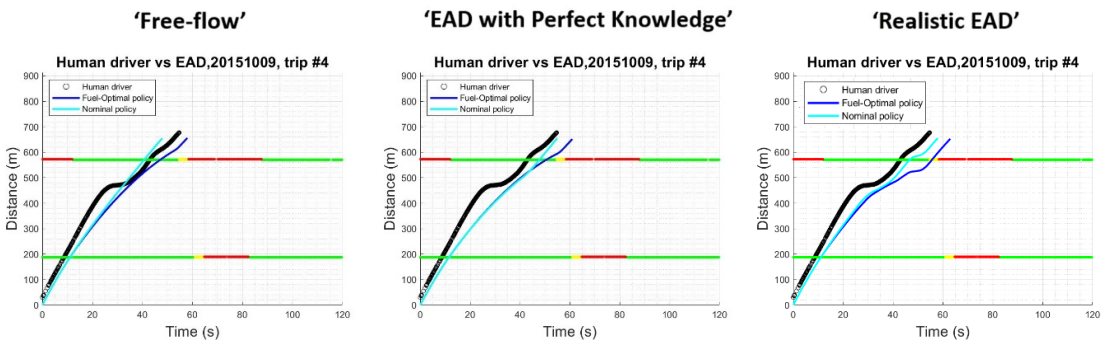
Figure 4.8. Two exemplar cases of realistic EAD problem. The trips were sampled from naturalistic human driving database and depicted in black. The proposed realistic EAD method first predicts the trajectory of the front vehicle as described in the left plots. Based on the predicted trajectories, the realistic EAD algorithm produces two optimal EAD trajectories, the fuel-optimal and the nominal policy as shown in the right plots. EAD algorithm solve alternates the two steps until the vehicle reaches to the goal [4].

The top case of Figure 4.8 is a representative case when instantaneous predictions at $t = 0$ worked well for $\forall t$. In this case, re-planning frequency barely affects the performance of EAD. On the other hand, the bottom case represents instances when unexpected events occurred which affected the motion of the front vehicle greatly and made the vehicle stop in the middle of the road. In this case, an instantaneous prediction $\hat{d}(t|t_0 = 0)$ will deviate greatly from the true $d(t)$ as t increases. The iterative prediction scheme resolves this problem. As depicted in the bottom left plot of Figure 4.8, when the re-plan frequency is short enough, the difference between predicted $\hat{d}(t|t_k)$ and the true $d(t)$ is bounded [4].

In order to evaluate how good realistic EAD performs compared to other EAD scenarios, i.e., free-flow scenario, we reproduced the same problem and solved EAD problem under other scenarios.



	Human	Free-flow		Perfect		Realistic	
Fuel	25.8g	10.6g	-59%	11.5g	-55%	11.6g	-55%
Time	68.6s	62s	-10%	70s	2%	71s	3%



	Human	Free-flow		Perfect		Realistic	
Fuel	25.9g	6.4g	-75%	9.2g	-64%	16.3g	-38%
Time	54.6s	48s	-12%	55s	1%	58s	6%

Figure 4.9. The comparison of the solutions of three different EAD scenarios with the same conditions. The left plots depict the unconstrained free-flow EAD solutions, and the right plots depict the realistic EAD solutions considering the trajectory of the front vehicle. The plots in the middle describe EAD solutions assuming the future trajectory of the front vehicle is known [4].

The figure 4.9 shows how the presence of the front vehicle impacts the fuel and time saving of EAD solutions compared to those of human drivers. The top three plots represent the solutions of the problem depicted in the top of Figure 4.8, where the realistic EAD solutions is almost as good as the free-flow solutions. The bottom three plots represent the solutions of the problem depicted in the bottom of Figure 4.8, where the realistic EAD solutions only perform half as good as the free-flow solutions. However, it was still able to save 38% fuel compared to human drivers while sacrificing only 6% of travel time. The center plots are the result of the scenario where we have full knowledge on the future states of the front vehicle. They provide insight on importance of accurate predictions and suggest the upper bound of fuel savings in realistic EAD scenarios.

Conclusion and future works

The free-flow EAD results identified the upper bound of fuel and travel time saving potentials, thus can serve as an upper bound of fuel saving potential of the eco-driving in the vicinity of signalized intersections. Realistic EAD results show the estimates in fuel and time saving of EAD when EAD techniques are implemented in practice. Note that the results only stand for the predefined scenarios of urban driving, where the vehicles go through a series of signalized intersections. The results do not represent the fuel and travel time saving of other driving scenarios in urban cities.

Future works include development of real-time prediction algorithm and corresponding real-time EAD algorithm. Note that the current algorithm usually takes longer than 1 minute to find an EAD solution. The future works also include identification of impacts of EAD on the surrounding traffics. The EAD algorithms will be implemented in traffic simulator to study how EAD vehicles impact macro-level traffic, including impact on the fuel and travel time of the surrounding vehicles.

References (For Task 4.4.2)

[1] D. Schrank, B. Eisele, T. Lomax, and J. Bak, 2015 Urban mobility scorecard, Texas A&M Transp. Inst., College Station, TX, USA, 2015.

[2] D. Bezzina and J. R. Sayer, Safety pilot: Model deployment test conductor team report, Nat. Highway Traffic Safety Admin. (NHTSA), Washington, DC, USA, 2014

[3] Geunseob Oh, Huei Peng, "Eco-driving at Signalized Intersections: What is Possible in the Real-World?", The 21st IEEE International Conference on Intelligent Transportation Systems, 2018

[4] Geunseob Oh, Huei Peng, "Extending Eco-driving at Signalized Intersections to Uncertain Urban Driving Environments", IEEE Transactions on Intelligent Transportation Systems (In preparation)

[5] Papoulis, Athanasios, and S. Unnikrishna Pillai. Probability, random variables, and stochastic processes. Tata McGraw-Hill Education, 2002: 148

Subtask 4.5 Implement and calibrate the POLARIS-Autonomie model

1 Introduction

The ultimate purpose of this task was to estimate the energy and mobility impacts under various traffic scenarios, diverse vehicle technologies and routes in order to assist people in optimal decision making in terms of mobility. This framework puts the energy criteria as one of the constraints to the vehicle and route choice, a dimension that has been lacking ever since in Polaris.

We currently have in effect a learning framework to predict energy consumptions for a variety of powertrains from cycle features or from more general route and trip features. The original work accounted for a training set that was representative of a population driving under standard cycle conditions. In this setting energy results were the product of pure physics based modeling and simulations. The simulation results were Autonomie generated.

In this subtask, the intent was to build on the existing machine learning based modeling developed and extend the decision rule scope by accounting for Real World Driving Behavior. We have in hand a rich set of real world data of drivers from the Detroit area that has the particularity of containing real time collected fuel points from On Board Diagnostics. The following is a walk through the data description to the modeling and results.

2 Data Description and Wrangling

The data is in the following format: mmddyy Oneday sample raw data.csv.

The data columns are:

VehId, Trip, TripStartUTC, TripEndUTC, Distance(mi), Timestamp(ms), Altitude[m], Latitude[deg], Longitude[deg], Vehicle Speed[km/h], MAF[g/sec], Engine RPM[RPM], Abs Throttle Position[%], Absolute Load[%], OAT[DegC], Fuel Rate[L/hr], HDOP, NumberOfSatellites, Accelerator Pedal Position[%], Air Conditioning Power[kW], Air Conditioning Power[Watts], Heater Power[Watts], Brake Pedal Position[%], HV Battery Current[A], HV Battery SOC[%], HV Battery Voltage[V], Odometer[kilometers], C2 Input Voltage[V], Is Driving[bool], Is Charging[bool], Fuel Rate Calc (g/s).

Random portions of the start and the end of each trip have been removed to protect PII information. The values of TripStartUTC, TripEndUTC, and Distance(mi) are unique to VehId and Trip. Other fields are updated at the rate described in figure 4.10. All fields EXCEPT Fuel Rate Calc (g/s) field are direct OBD2 outputs. Fuel Rate Calc (g/s) is the fuel consumption estimate calculated from Mass Air Flow(MAF).

Not all the columns are useful for a machine learning based model, the next sections will go through the feature selection process.

The data consists of 10722029 observations of 42 variables. This file is used as an input to the map matching algorithm, mapping every vehicle drive cycle onto the detroit area route network. This corresponds to 3 month worth of data. Information such as route length, route speed limit and other will be useful as additional features to a machine learning based model.

Data Name			Data Type	Populated %	Update Period
GPS	Latitude/Longitude (deg)		Float	94.2	Every 5 (sec)
	Altitude (m)		Real	97.2	
	HDOP		Float	97.4	
	Number of Satellites		Tinyint	97.4	
Vehicle Speed (km/h)			TinyInt/Float	90.6	1 (sec)
Fuel Info	Engine RPM (rev/min)		Int	90.1	2 (sec)
	Absolute Throttle Position (%)		Float	2.3	
	Mass Air Flow (g/s)		Float	72.0	
	Fuel Rate (L/hr)		Float	1.5	5 (sec)
	Absolute Load (%)		Float	74.8	
PHEV & EV only	Odometer (km)		Float	27.1	30 (sec)
	Ambient Temp ©		Float	9.6	60 (sec)
	Auxiliary Power (HVAC)	AirCon Power (KW)	Float	36.7	
		AirCon Power (Watt)	Int	56.8	
		Heater Power (Watt)	Int	20.5	
	Battery SOC (%)		Float	96.1	
	Battery Voltage (V)		Float	88.9	5 (sec)
	Battery Current (A)		Float	88.9	
	Is Driving, Charging (bool)		TinyInt	22.8/63.0	-

Figure 4.10: Data collection rate

3 Map Matching

What is the map matching algorithm?

The purpose of map matching is to connect recorded GPS trip points to the links of road networks and establish trip routes. Extensive GPS-based trip data processing requires both accuracy and computational efficiency of the map matching code. Generally, map matching algorithms make use of geometric methods, topological or probabilistic models for accuracy at the expense of a very slow process.

Why make the connection?

By constructing trips from GPS data we are able to calibrate an existed agent-based transportation model of the city of interest.

What is the problem?

In the previous section we noted that for only 3 month of data collected at rates close to 1-10Hz that were shrank onto summary statistic quantities resulted in over 10 million points. This is an immer- sive amount of GPS data collected from high-accuracy in-vehicle GPS. Without shrinkage, a regular trip data set collected can amount to 50, 000, 000 points, and processing this large amount of data could take several weeks. Therefore calibrating an agent based model over a regular trip is unreasonable.

Solution

The developed algorithm [LAS15] is based on Multiple Hypothesis Technique, which was firstly introduced by [PSS01]. Several routes candidates are kept in following the sequence of GPS points, developed and scored to find the best candidate, and only determined the best path at the end of the sequence. [MHA05] adopted a topological search algorithm which proved to be more efficient for larger scale map matching problems. By limiting the number of candidates kept in memory, computational feasibility is guaranteed.

How does it work?

The details of the map matching algorithm has thoroughly explained in [LAS15] The overall workflow is:

1. Pre-process: trimming out irrelevant points, reduce the number of processing points, and grouping points to each person's trips.
2. Initialization: determining the starting links and assign routes to each candidate links.
3. Development: tracing the points sequence to add new connected links for each route.
4. Reduction: scoring each route when adding new routes to the pool, and remove routes with worst performance to save memory.
5. Post-process: trimming out irregular links in candidate routes through topological criteria.
6. Selection: selecting the best route as the final result

The map matching process results in a large dataset of vehicles mapped to the Detroit roads. Each vehicle and trip is assigned a person id a trip id and a link id A trip is a set of aggregated links from which several parameters are known from the GPS information and the Detroit Map. In particular, for each link we know the length of the link length link, the length of the actual distance covered by the vehicle on the link length gps, the amount of time spend by the vehicle of the link time spent, the amount of time the vehicle stopped on the link time stop, the average speed of the vehicle on the link speed avg, the speed limit on the link speed limit, a Boolean indicating whether the vehicle covered.

The entire/most of the distance for the link, the actual coverage percentage coverage which is computed from known length link and length gps, and finally the amount of fuel consumed on the link fuel. key is a tracking grouping key for vehicles and trips. Distances are in meters, Times in seconds, Speeds in meters per second and Fuel amounts in g/s, other quantities are unit free.

4 Add more attributes

Polaris is a macro simulation software and the amount of information on each link is quite limited. In this subsection we show how from the attributes generated by the map matching algorithm we can compute extra route related features that may or may not be useful to explaining the variability in the fuel. The section related to the model and feature selection will walk through which of those attributes is useful to any model.

A new database is then constructed with new attributes: length link p and length link n which are respectively the immediate length of the previous and next link. length gps p and length gps n which are respectively the immediate actual length covered by the vehicle on the previous and next link. Intuitively, those attributes may not be very useful as the length of the previous or next link do not seem to interfere with the current link fuel consumption. The feature selection and analysis section will take care of making the decision for us based on a serious analysis.

Also added attributes are speed avg p and speed avg n which are respectively the immediate average speeds of the previous and next link, speed diff p and speed diff n which are variables describing the differences in speed between the current link and respectively the previous and next link. Finally we also compute a traffic load variable which, due to the lack of data, will be used as a proxy to traffic conditions within the link. This is typically calculated by taking the ratio of speed avg/speed limit. Other features can be also constructed as an indication to traffic conditions within the link, for example computing the ratio time stop/time spent can be of interest however we decide to skip it in this first path. In the future, we also could consider the effect of adding vehicle speeds of not only the links of direct proximity but extend to links that are further away as well. This introduces a time correlation between the links and their property and autoregressive techniques can be used in this case.

5 Maps

The ability to visualize data with geographic context is a valuable exercise to do. It can give us a sense of what is happening behind the numbers. One of the main issues that our map matching algorithm encounters is that several vehicles and trips go out of range, the boundaries here being the area of Detroit for which a Polaris map is provided.

As it can be seen from the first set of plots in Figure 4.11, many vehicles go beyond Detroit area. Person 5 had one trip to Toronto, Canada. Person 255 and 362 had at least more than two long range trips, to Ohio, Pennsylvania and other. Person 2 had only one long trip, the rest in the city of Ann Arbor as we will see later. Unfortunately this data will be scrapped. This is valuable information that won't be used. We clarify here that the colors represent different trips for a same driver.

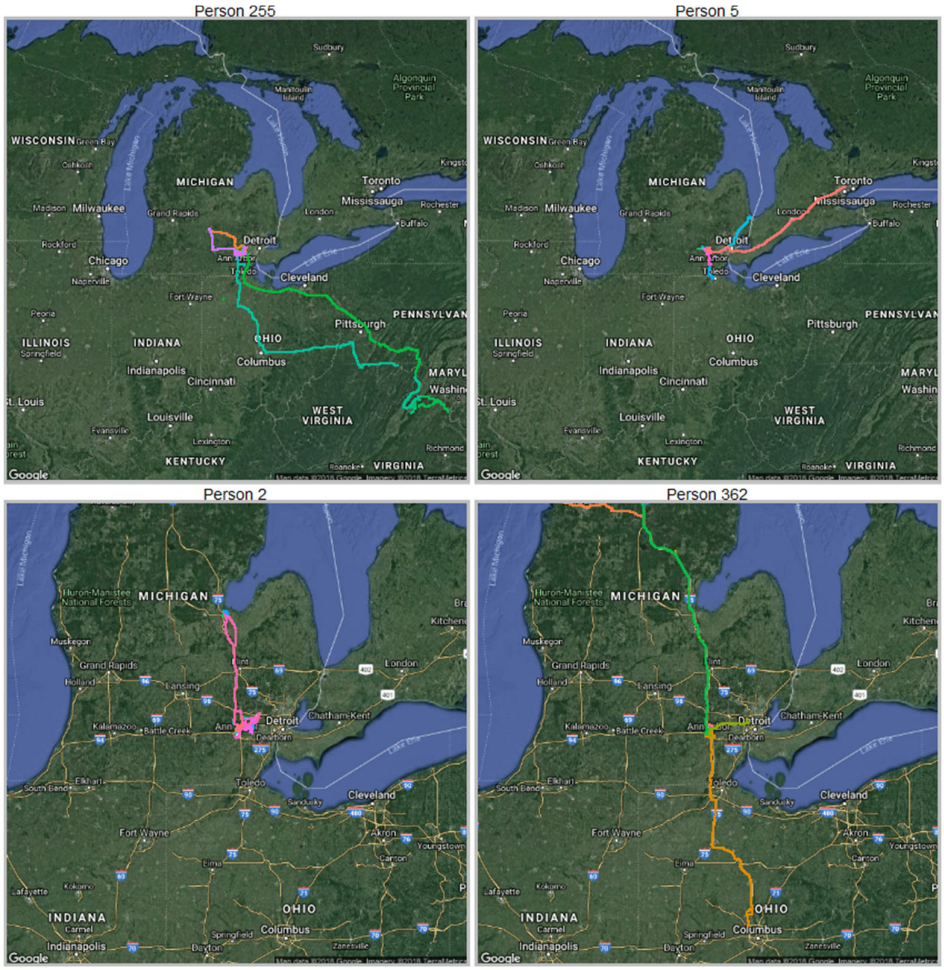


Figure 4.11: Example of long out of Detroit range trips

The next plots in Figure 4.12 show examples of inbound drives, particularly in the city of Ann Arbor and its surrounding. We see here that Person 211 is a good example of trips that can be categorized as highway driving, most the trips from this person circle Ann Arbor via its highway roads and go beyond. On the flip side Person 228 seem to have a balanced combination of city and highway driving and some of its trip go to downtown Ann Arbor.

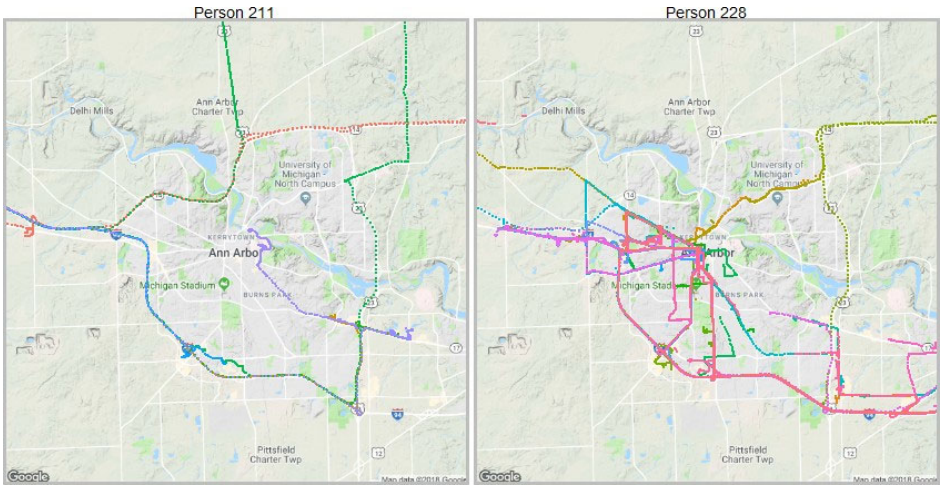


Figure 4.12: Example of long out of Detroit range trips

Next in Figure 4.13 we illustrate examples of pure in city drives, in particular, in the downtown area of Ann Arbor.

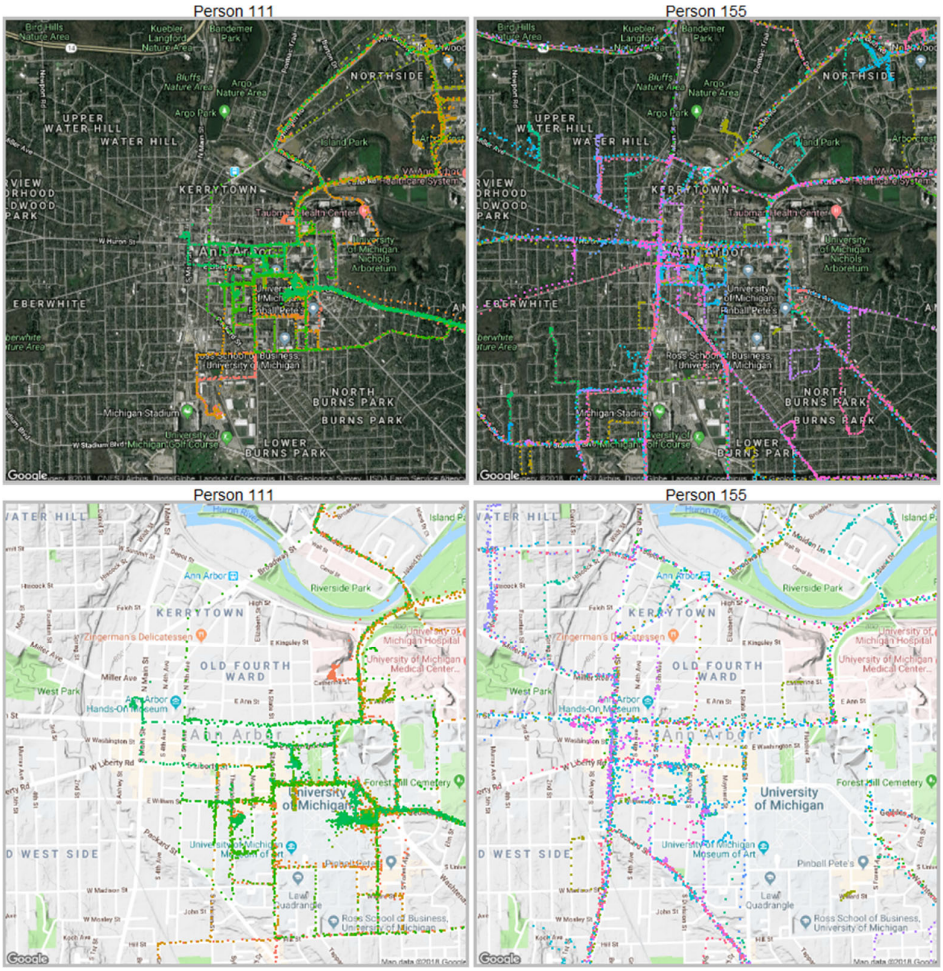


Figure 4.13: Overview of downtown Ann Arbor for 2 drivers over multiple trips.

Subtask 4.6 Model validation with new data from field tests

We build a first model and make use of it for outlier detection purposes. The model summary suggests that several predictors are not significant to explaining the variability in the response. We will not worry about this now and focus and detecting outliers that may affect the model structure and prediction power. An upcoming section will deal with the model selection problem. First we need to check the independence, constant variance and normality of the error. Keep in mind this is not a final model for prediction but only a model constructed for the purpose of detecting outliers. The final prediction model is based on neural network principles and is detailed in a future section. Even so, we note that the residual standard error for this model is of the order of 0.0063 this is relatively good and likely to yield acceptable predictions. As a comparison below is a summary statistic of the response variable fuel.

Figure 4.14 shows a boxplot and a violin plot for the response variable. Note the width of the boxes is directly proportional to the quantity of data behind it, this is a very important feature as boxplots hide the distribution and number of points behind it. The violin plot gives an additional flavor to the distribution of the data. It is used to visualise the distribution of the data and its probability density. Here the boxes are broken down by vehicle class. The original data is extremely skewed, right tailed, the median fuel value averages (for some classes) at around e−5 = 0.0067, this is quite close to the residual standard error of the model. In other words, about 50% of the fuel points could be predicted quite accurately.

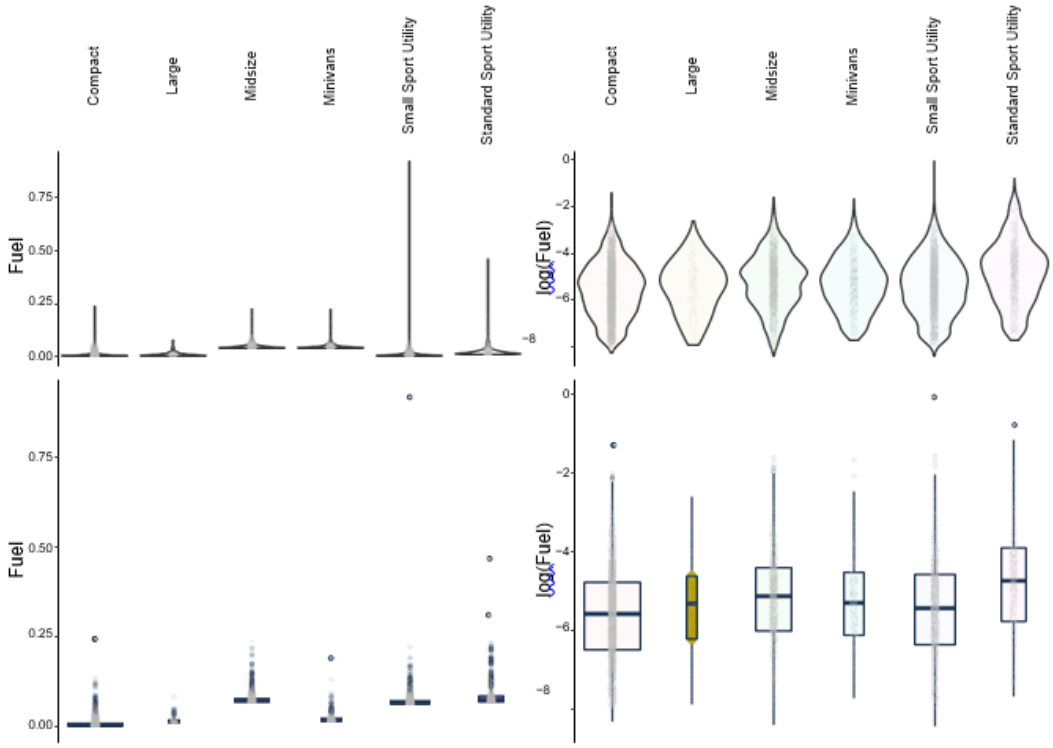


Figure 4.14: original skewed (left) and log transformed (right) fuel data

We run a series of diagnostic plots to check several of the assumptions. The residual vs. fitted plot shown in Figure 4.15 is of the most useful ones, it is noted that there is some clear sign of heteroscedasticity in the vertical direction. This is an indication that some modifications need to be done in the model towards high fitted values. High response values (which are linked to high leverage points) seem to behave differently, it may be worthwhile to fit a different model for high response

values and have some kind of broken stick regression or apply some kind of transformation. A Box-cox transform can help get rid of the issue. This is expected as we saw that the fuel is heavily right skewed.

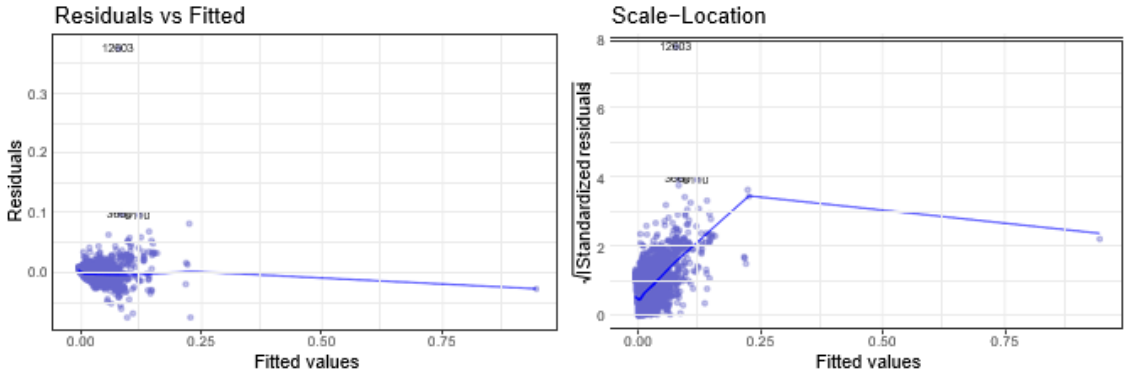


Figure 4.15: Residual plots

As matter of fact a histogram of the fuel data shows in Figure 39 that a log-normal type of model or a generalized linear model with a log-link transform could be adequate. Since a neural network model will be used to fit the data anyway, we only briefly go into considering those kind of models next but do not get deep into the pain of making these models perfect; the main purpose of those models is do get a sense of the tendencies and relationships between the predictors and the response as well as detecting any kind of anomalies. We also remark a few extreme points for which are suspect to outlyingness and need some further investigation on their validity (discussed next). The scaled location plot is an effective way to increase the resolution of the residual by considering the absolute value of the residuals.

Subtask 4.7 Vehicle energy consumption quantification

Artificial neural network (NN) models are used for various purposes and in different fields. In recent years NN have been showing a large amount of hype, maybe because of the fancy name or probably also because of the idea behind it of mimicking biological neural networks and brain functions. In reality neural network models are just an additional methodology and is simply viewed as an algorithmic procedure that rivals other regression, classification and clustering methods used by statisticians. Although NN models and more generally deep learning showed some good promises, the success and hype surrounding deep learning is truly attributed to the specific convolutional neural network model architecture under unsupervised learning with big data applied to image recognition.

4.7.1 Data to Prediction Routine

The process of any machine learning procedure is practically standard. The flowchart in Figure 36 illustrates the usual steps taken from A to Z. In each step, the machine learning philosophy compel us to make sure that the ultimate purpose of each step is to serve the need of delivering a good predictive model. Unlike in statistical fields, the prediction power is the sole purpose of the resulting model, without a need to understand or interpret the relationships between the inputs and outputs. In this process, it is easy to lose track of the original values or the original meaning or the units of any variable. The very first steps of the flow diagram have already been accomplished and detailed in the previous sections, this comprise the identification of the data sources, the data collection, the data integration and the creation of a combined database coming from multiple sources. We also described how the data was prepared and we went through a preliminary exploration of the data to understand the nature of the data, the format and its quality. This step usually gives us a lot of insight and leads us to an informative analysis and increase the chances of a successful outcome. After, we presented how we performed many data manipulations as part of our preprocessing step. This by cleaning obvious erroneous points, selecting variables of interest from a correlation analysis and applied many transformation to suit our needs. This cleaned dataset was then passed on to a more elaborate analysis and for the purpose of outlier detection which we have detailed and feature selection. The final dataset is then ready and through the right analytical techniques an appropriate model can be built. The model

build includes a training and validation step in which model parameters are tuned and optimized and accuracy of predictive results controlled via cross validation methods to avoid overfitting. The results are then assessed and predictions can then be communicated. It is important to understand that machine learning routine is an iterative process, the training and modeling may need to be revisited many times.

We have been working on the development of a prediction tool Figure 4.16 to ease visualize predictions. The tool is intended to provide the user with a generic data-independent interface and guide him to create, configure, train and view machine learning models. It also allows users to visualize and evaluate model performance. In addition, the tool also offers an automated outlier detection process based on Random sample consensus methods (RANSAC). This feature is useful when data need to be fit automatically without the intervention of a skilled analyst. Although useful, it is dangerous to exclude outliers in an automatic way, and one need to be aware of this. The tool contains two several modes: a real time and a batch mode. The real time mode is designed to make quick predictions, visualize the results and analyze them while quickly be able to modify route, trip or vehicle inputs. The batch mode allows the user to define a large set of inputs at a time, prediction is performed on all and a output prediction file is generated. The latter method is appropriate to study different scenarios and analyze the behavior of various vehicles or trips in different settings.

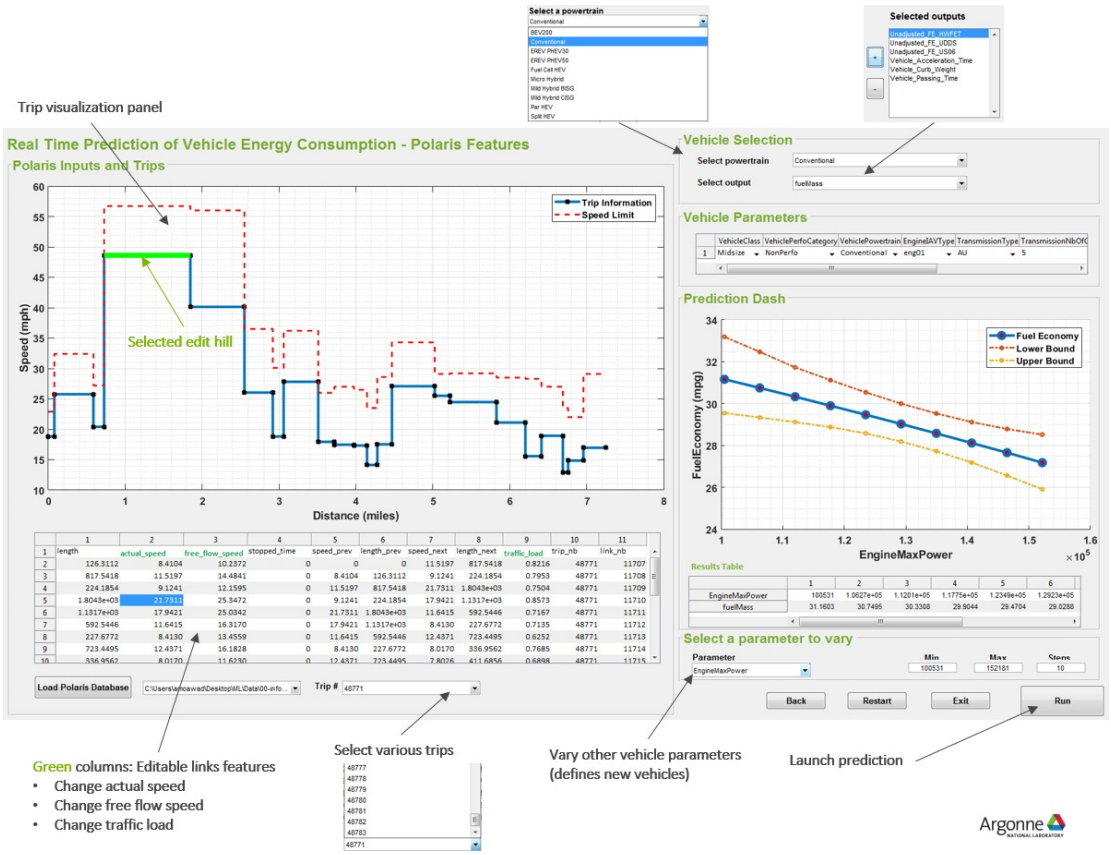


Figure 4.16: Machine Learning based Prediction Tool

We have discussed the limitation of the model developed and its prediction accuracy. We have also detailed the conditions in which the accuracy can be hindered. Figure 4.17 shows an example of a relatively bad prediction. We see in this example that all the links are short very short in length and the driver does not spent enough time on each. Half of the cycle can be considered as a city drive

(below 55 mph). The resulting prediction for this vehicle on the selected trip exhibit a 12% error. This can be considered unsatisfactory to some extent.

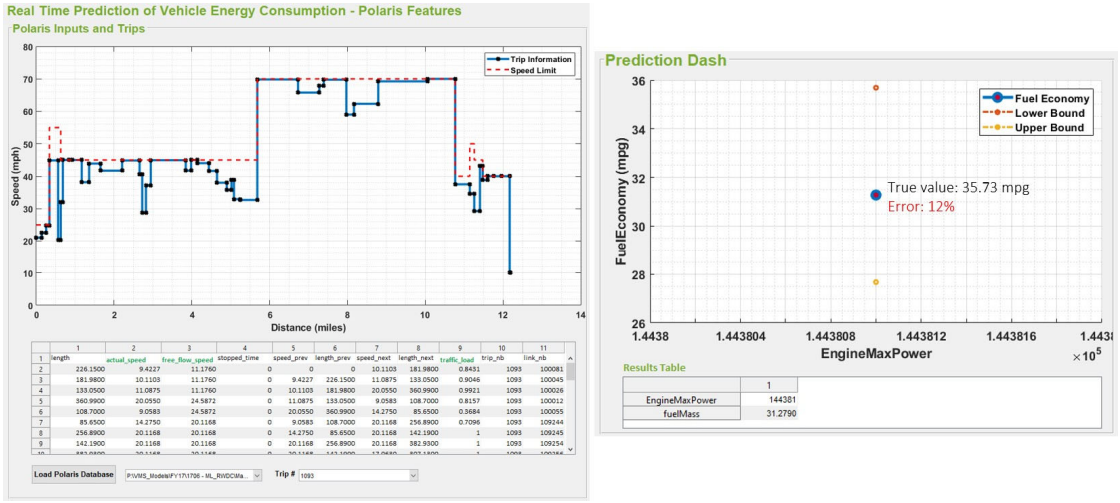


Figure 4.17: Example of a bad prediction

Task 5 Adaptive Signal Control

In this task, we investigate the energy savings and mobility benefits from adaptive traffic signal control systems with varying percentage of connected and automated vehicles (CAV). There are four sub-tasks. First, a hardware-in-the-loop (HIL) simulation environment is built and calibrated, which setup a virtual testing environment (Task 5.1). Second, five algorithms are developed for different CAV penetration rates under mixed traffic conditions (Task 5.2). The proposed algorithms are tested and validated in the HIL simulation environment in terms of both mobility and energy benefits (Task 5.4). Finally, an implementation case study is developed to conduct field experiment along a real-world corridor (Task 5.3).

Table 5.1 shows the content and schedule of each sub-task.

Table 5.1 Content and Schedule of Each Sub-task

Subtask	Content	Schedule
5.1	Build and calibrate the traffic simulation environment	M1-M6
5.2	Develop the adaptive signal control algorithm	M7-M18
5.3	Deploy and conduct field experiment at Plymouth Rd	M19-M30
5.4	Evaluate the energy saving of adaptive signal control	M31-M36

5.1 Traffic Simulation Environment

5.1.1 Simulation Platform Overview

A hardware-in-the-loop (HIL) simulation platform is designed to test and evaluate the models in a microscopic simulation environment. The simulation platform aims to replicate the real-world situation as much as possible so that models and algorithms tested in simulation can be deployed in the filed with minimal modification. The structure of the simulation platform is shown in Figure 5.1.

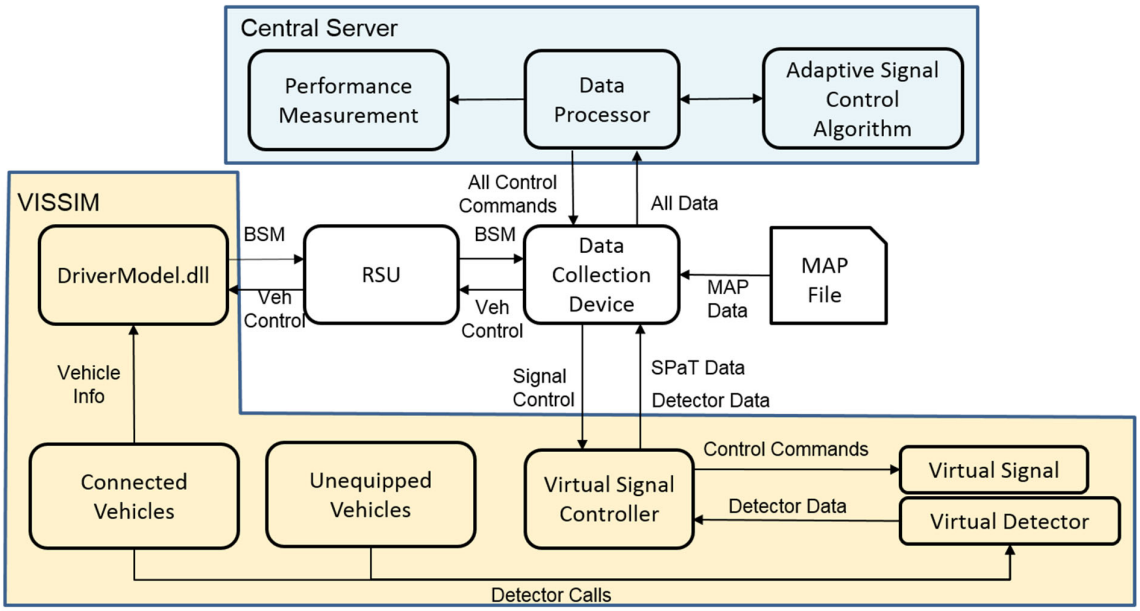


Figure 5.1: HIL Simulation Platform

The HIL simulation platform is designed for a CAV environment. There are mainly three parts: a simulation software, a central server and two pieces of hardware device: a roadside unit (RSU) and a data collection device.

VISSIM which is a microscopic simulation software is selected to simulate vehicle movements and traffic signal operations. VISSIM is able to simulate individual vehicle behaviors such as car following and lane changing as well as different types of traffic signal operations including fixed, actuated and adaptive. VISSIM allows users to control all or part of the vehicles and change their behaviors (e.g. Eco-departure, Eco-approaching) based on user defined models. The DriverModel.dll API is used to generate Basic Safety Message (BSM) for each CAV in VISSIM. Each BSM includes the real-time basic vehicle information including vehicle ID, location, speed, heading, and acceleration etc. The transmission frequency of BSM is 10Hz. The BSMs will be sent to RSU first and then the RSU will forward the BSMs to the data collection device.

The signal phasing and timing (SPaT) data is sent out by the virtual signal controller in VISSIM to the data collection device every 0.1 second including current vehicle phase status, pedestrian phase status, overlap phase status, and estimated remaining time of each phase.

A map description file which describes the geometric structure of the intersection is a static file locally stored in the data collection device. This file includes the GID information such as GPS coordinates of the lane nodes, lane attributes (e.g. allowed movements) and lane to lane connection etc.

All data including BSM, SPaT and map will be sent to the central server which has three components: data processor, performance measurement algorithm and adaptive control algorithm. The data processor is developed to store and process all data from the data collection device for algorithm development. The adaptive control algorithm is designed to generate optimal signal timing plans with energy and environmental objectives such as minimization of fuel consumption and emissions. The performance measurement algorithm is used to evaluate the performance of the adaptive control

algorithm. After a new signal timing plan is generated, this new plan will be executed through sending National Transportation Communications for ITS Protocol (NTCIP) commands to the virtual signal controllers in VISSIM.

5.1.2 VISSIM Simulation Model

A VISSIM (PTV, 2013) simulation model is built for the six-intersection corridor at Plymouth Rd, Ann Arbor as shown in Figure 5.2. The six intersections are: Plymouth Rd @ Green Rd, Plymouth Rd @ Huron Pkwy, Plymouth Rd@ Nixon Rd, Plymouth Rd @ Traverwood Dr., Plymouth Rd @ Murfin Ave, and Plymouth Rd @ Barton Dr. The Plymouth Rd has two lanes for each direction which is one of the busiest commuting route, serving US23 to the North campus of UM and downtown Ann Arbor. Some crossing roadways are major arterials which carry large volume of traffic (e.g. Green and Huron) and others are side streets with less traffic demand (e.g. Traverwood Dr.). The road geometries are calibrated with the satellite maps from Google Earth.

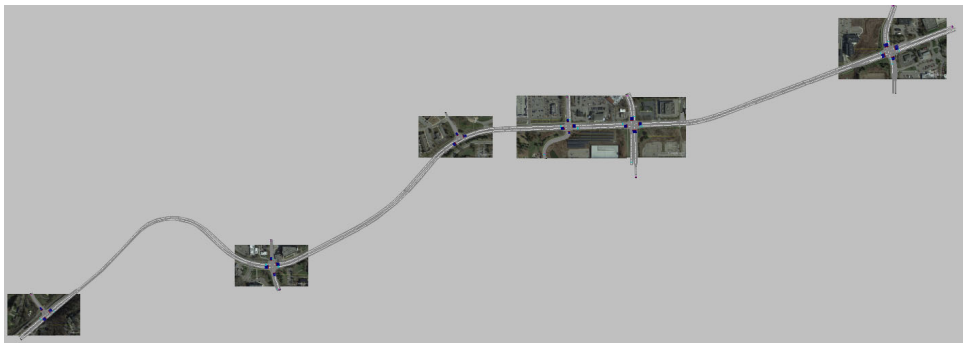


Figure 5.2: VISSIM Simulation Model of Plymouth Corridor

Figure 5.3 shows vehicles, traffic signals and stop-bar detector layouts of the intersection Plymouth Rd and Huron Pkwy. The blue vehicles in the simulation are regular vehicles while the red vehicles are CAVs. Note that only CAVs broadcast BSMs. The market penetration rates of CAVs can be modified by setting up different vehicle compositions.

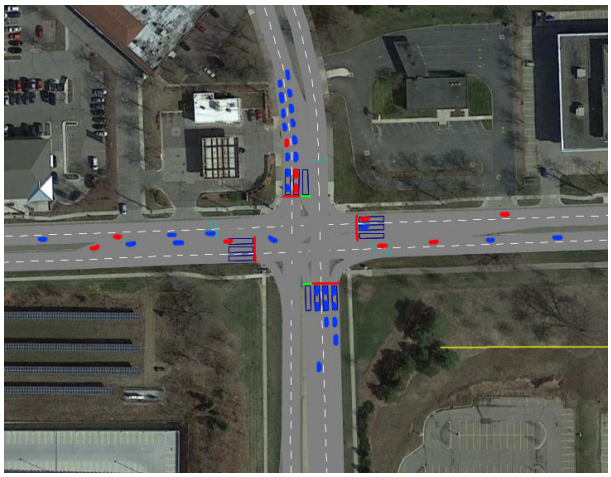


Figure 5.3: Plymouth Rd and Huron Pkwy Intersection in VISSIM

5.1.3 DriverModel.dll API

The drivermodel.dll API is an interface to VISSIM that allows users to apply different driving behavior models for some or all vehicles in VISSIM. The API is implemented as a dynamic link library (DLL) written in C/C++. The API provides several functions to create, move and delete vehicles as well as read and set vehicle parameters for car-following and lane-changing models. Any vehicle type (e.g. CAV) can be

enabled to call the drivermodel.dll every simulation step through the vehicle type property page as shown in Figure 5.4.

In this project, the DriverModel.dll is used to generate BSMs as mentioned above. The process of generating BSM using the API is described below:

Step 1: Initialization: Setup UDP socket communication and read IP address and port of the target RSU

Step 2: Read vehicle information from VISSIM through DriverModelSetValue() function

Step 3: Coordinates transformation which transform the vehicle position coordinates from local X, Y to GPS coordinates (WGS-84) applying the transformation algorithm described in (Farrell and Barth, 1999). This algorithm first transforms local X, Y coordinates to the earth-centered earth-fixed (ECEF) rectangular coordinates. The ECEF coordinates has its *x* axis extended through the intersection of the prime meridian (0° longitude) and the equator (0° latitude). The *z* axis extends through the true North Pole. The *y* axis completes the right-handed coordinate system, passing through the equator and 90° longitude. Then the ECEF coordinates are transformed to GPS coordinates. The relationship among the three coordinate systems is shown in Figure 5.5.

Step 4: Generate Society of Automotive Engineers (SAE) J2735 standard BSM using an open source ASN.1 encoder/decoder (http://lionet.info/asn1c/compiler.html).

Step 5: Broadcast BSMs through the UDP socket.

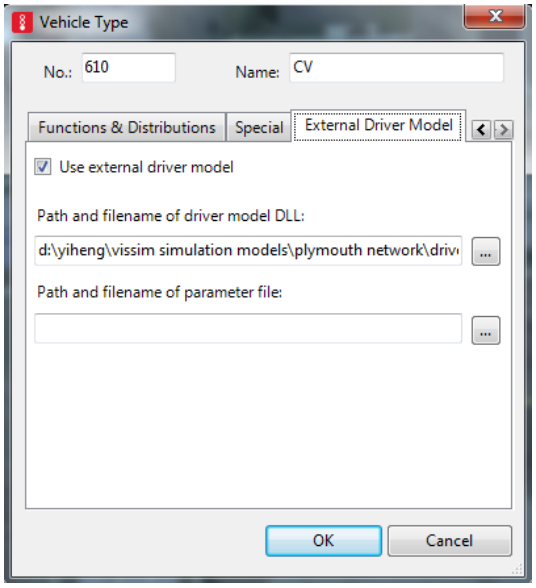


Figure 5.4: Vehicle Property Page for External Driver Model

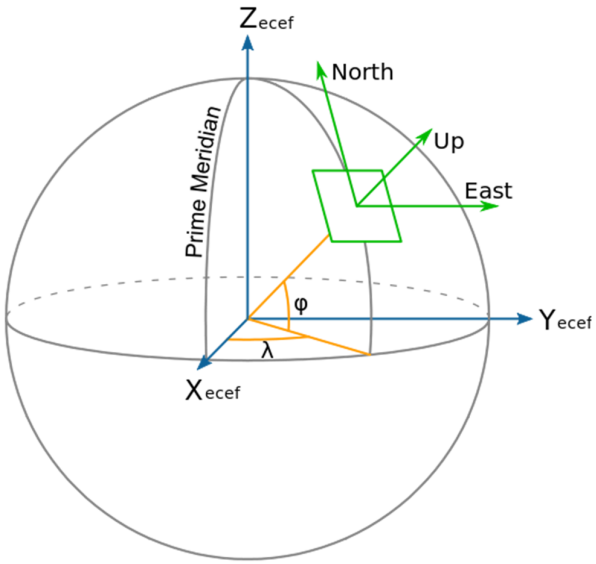


Figure 5.5: Relationship Among Local, ECEF and GPS Coordinates Systems

Source: Digital Imaging and Remote Sensing Lab (http://www.dirsig.org/docs/new/coordinates.html)

VISSIM is running in real-time and the simulation resolution is set to be 10 steps per simulation second. As a result, every 0.1s of simulation time (also 0.1s of actual time), BSMs from all CAVs are broadcast to the RSU which is the same frequency as the field operation.

5.1.4 Data Collection Device

The data collection device is designed to interface with both roadside unit (RSU) and traffic signal controller to simultaneously collect DSRC messages from connected vehicles, as well as high resolution detector and signal status data from traffic signal controller. The device can also encode signal status and road geometry information into standard DSRC messages. These messages are sent to RSU to be broadcast in real-time. The data collection device pre-processes the data and sends to the central server for storage, management and visualization as shown in Figure 5.6.

The device is aimed to be developed as a universal interface device, which is compatible with various types of RSUs, traffic signal cabinets and controllers. This device is independent of controller and RSU vendors, and can serve as a cost-effective way to upgrade existing infrastructure without much changes to other devices.

The RSU receives BSMs from all connected vehicles within the DSRC communication range and forwards to the data collection device. The device receives SPaT data from the signal controller and MAP data from a local description file and generates SAE J2735 SPaT and MAP messages and then forwards to the RSU. The data collection device also receives the loop detector data and signal status data from harness cables or the Synchronous Data Link Control (SDLC) port. It preprocesses all the data including information extraction, data reduction and data re-formatting and sends to the central server with a predefined format.

Figure 5.6: The Data collection Device

5.1.5 Central Server

The central server is designed to process and visualize data from different intersections (data collection devices) as well as generate optimal adaptive signal timing plans regarding energy efficiency. It has three components: a data processor, a performance measurement component and an adaptive signal control component.

The data processor categorizes received data and saves to different databases. In addition, the data processor calculates additional vehicle and intersection information based on received BSMs and MAPs. For example, an algorithm is developed to locate vehicles on the roadway based on the intersection map and the vehicle's GPS location from BSM. Additional information including vehicle current approach, current lane, distance to stop bar, requested signal phase and vehicle states (e.g. approaching, leaving or in queue) are generated from the algorithm, which are critical to traffic signal control applications.

The performance measurement component reads data from the databases and evaluate current traffic conditions such as traffic volume, queue length, travel time and level of service (LOS) at different levels including intersection level, corridor level and network level.

5.2 Adaptive Signal Control Algorithms

The overall algorithm development framework is shown in Figure 5.7. The algorithms are divided into two branches: with connected and automated vehicles (CAVs) or with connected vehicles (CVs). CVs refer to those vehicles that are able to communicate with each other and the infrastructure. However, they are still driven by human drivers. As results, CVs are only observable but not controllable. CAVs refers to those vehicles that have both connectivity and automated driving functions that can be controlled by computers.

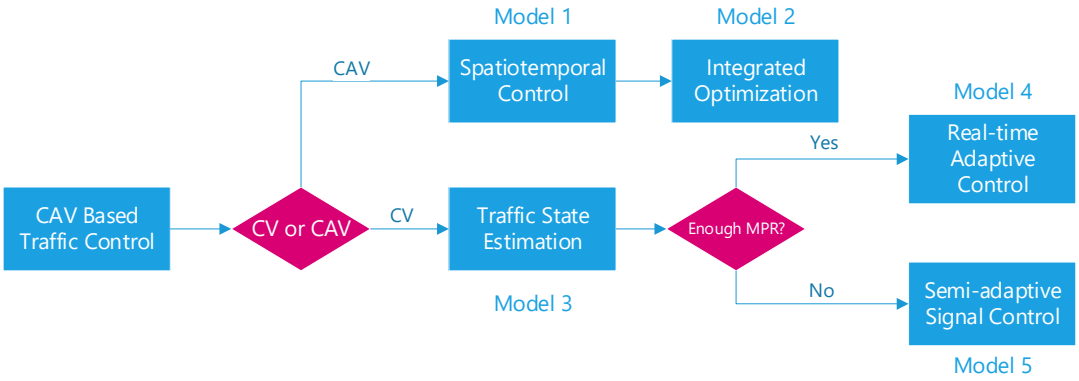


Figure 5.7 Algorithm Development Overview

The first branch assumes all vehicles are CVs and a portion of vehicles are CAVs. Then the traffic control framework can be expanded to two dimensions: spatial (vehicle trajectory control) and temporary (traffic signal control). Based on this concept, we developed a spatiotemporal intersection control framework that applies a two-stage optimization model to jointly control CAV trajectories and traffic signals (Model 1). Further we integrated the two-stage optimization model into a unified framework and proposed an integrated model to optimize signal timing, vehicle arrival times, and vehicle lane changing behaviors simultaneously (Model 2).

The second branch assumes only CVs exist on the road, which implies an early deployment stage. One of the major challenges at this stage is the low penetration rate problem. Only a small percentage of vehicles are connected so that an estimation model is necessary to estimate the entire traffic state based on limited CV information. To this end, we developed a model that estimates traffic volume based on CV trajectory data (Model 3). Traffic signal optimization can be conducted based on estimated volume. Depending on the penetration, two different optimization models are constructed. If the penetration rate is high enough (e.g., >10%) and sufficient number of real-time CVs are observed, a real-time adaptive signal control model was formulated (Model 4). If the penetration rate is extremely low (e.g., <5%), so that there is not enough real time information, a semi-adaptive signal control model was implemented (Model 5).

The following sub-sections introduce each of the model in more details.

5.2.1 Spatiotemporal Intersection Control

Current traffic signal control strategies including fixed-time, vehicle-actuated and adaptive control allocate green times to different vehicle movements to avoid conflicts and ensure intersection safety. With the rapid development of connected and automated vehicle (CAV) technologies, vehicles can communicate with the RSU through dedicated short range communications (DSRC). At the same time, data from the RSU (e.g. signal status and intersection map) can be broadcasted to vehicles within the communication range. The two-way real-time communication between the CAVs and the infrastructure makes the vehicles “controllable” through either speed advisory system for human-driven vehicles or control systems in connected and automated vehicles.

Therefore, in a CAV environment, not only traffic signals but also vehicle trajectories can be controlled to improve traffic efficiency and gain environmental benefits. Current research efforts mainly address only one side of the control problem. For example, Eco-driving (Barth et al., 2011; Rakha and Kamalanathsharma, 2011) and speed advisory (He et al., 2015; Wu et al., 2015) mainly focus on vehicle trajectory control with the purpose to reduce fuel consumption or emission. These applications assume that signal timing is fixed and known to the vehicles. Meanwhile, CAV based signal control applications (Feng et al., 2015; Goodall et al., 2013; He et al., 2014; Lee et al., 2013) consider vehicle

trajectories as the input to signal control algorithms. The objectives of signal optimization are usually only related to efficiency such as minimizing total vehicle delay or maximizing throughput. Energy impacts are seldom considered.

To the best of our knowledge, there are only quite limited studies on the joint optimization of vehicle trajectories and signal timing. (Malakorn and Park, 2010) proposes a cooperative system where a vehicle trajectory is assumed to include an acceleration segment and a cruising segment. Vehicle arrival windows are then calculated. On the basis of these windows, traffic signals are optimized. Simulation analysis justifies the system in terms of both mobility and environmental impact. (Li et al., 2014) proposes an algorithm to jointly optimize vehicle trajectories and traffic signals. A generic optimal vehicle trajectory of four segments is assumed. The first and third segments have constant acceleration/deceleration, while the second and the fourth segments have constant speed. If the time durations of some segments are zero, the actual trajectory may have less segments. Owing to the simple case in the research, signals are optimized by enumerating all feasible combinations of the number of phases and phase splits. Obviously, the assumption of an optimal vehicle trajectory with determined segments cannot guarantee the true optimal solution. The enumeration method may be ineffective when complex phase structure is considered. In addition, this method requires to control all vehicles which can only be applied in a fully automated environment.

One notable solution to the joint control problem of vehicle trajectories and traffic signals is so called "free" intersections where the traffic signals are removed and all vehicles pass the intersection in a self-organized way (Lee and Park, 2012; Zohdy and Rakha, 2014). However, this approach requires 100% penetration rate of fully automated vehicles, which is not realistic in the near future. Given the update rate of vehicles in the U.S, it can be predicted that in the next ten to twenty years, traffic signals will still play an important role in urban transportation operations.

Model Framework

The proposed joint control framework aims at improving the efficiency of green time utilization to minimize vehicle delay and smoothing vehicle trajectories to reduce fuel consumption and emission. Figure 5.8 shows the comparison of green time utilization between the state-of-practice signal control and the proposed joint control.

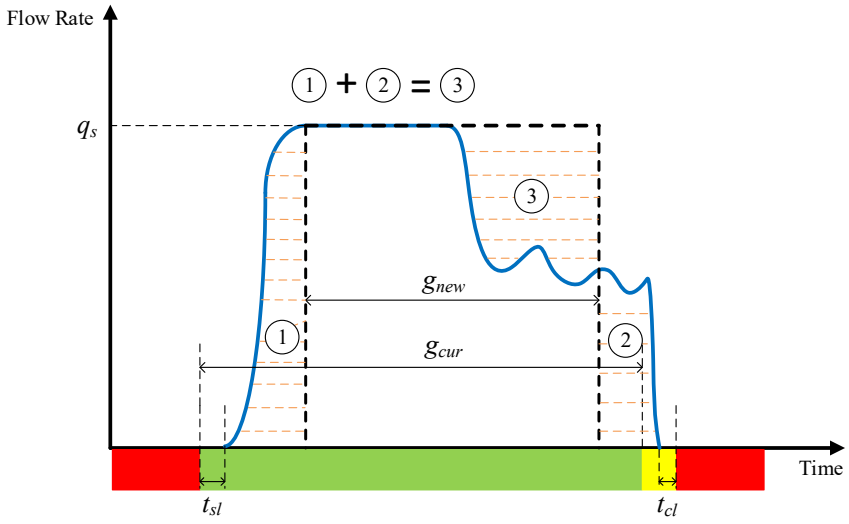


Figure5.8: Green Time Utilization Comparison

The blue curve shows the vehicle discharging rate at an intersection under current signal control strategies. When the signal turns to green, the first few seconds of the green time are wasted (start-

up lost time t_{sl}) because human drivers need time to respond to signal changes and accelerate the vehicles. Then the discharging rate starts to increase to the saturation flow rate (q_s) until queued vehicles are fully discharged. Finally, it drops to the arrival rate which is usually lower than the saturation flow rate. The area below the curve is the vehicle demand served during the green time. In the proposed framework, the trajectory of the leading vehicle of an approaching platoon is controlled so that it arrives at the intersection at the beginning of the green time with an optimal speed. The control of the leading vehicle trajectory also results in a compact platoon so that the discharging rate keeps the saturation flow rate (as shown in the black dashed line). If Area 3 is equal to the summation of Area 1 and Area 2, then within much shorter green time interval (g_{new}), the same number of vehicles can pass the intersection as that in the current control strategy where the green time is much longer (g_{cur}). As a result, the green time utilization is greatly improved. Note that the proposed framework doesn't increase intersection capacity by shortening the saturation headway, but by utilizing green time more efficiently.

To achieve this goal, a two-stage optimization model is proposed. In the first stage, adaptive signal control concept is adopted to address the flow fluctuation. Total vehicle delay is used as the objective function (1):

$$\min_{g_p, u_p} D = \sum D_p(g_p, u_p) = \sum_p \sum_i d_{p,i}(g_p, u_p) \tag{1}$$

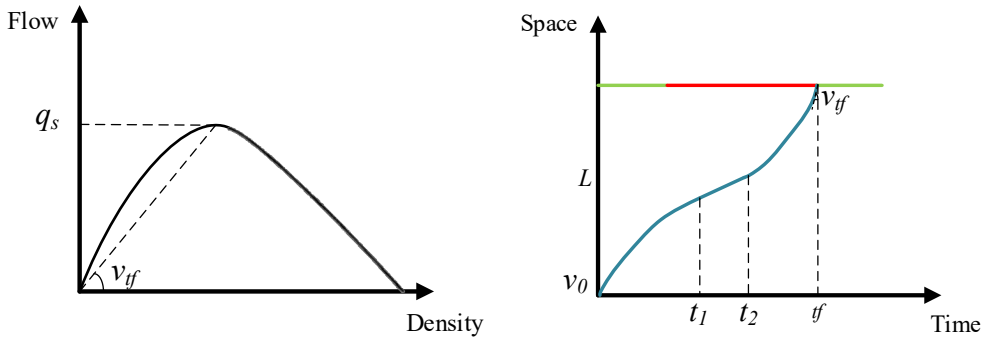
$$\text{s.t. } \boldsymbol{g}(g_p, u_p) \leq 0 \tag{2}$$

where $d_{p,i}$ is the delay of vehicle i in phase p; g_p is the remaining green time of phase p ; u_p is the acceleration/deceleration rate profile of the leading vehicle in phase p. u_p, given the value of g_p, is the solution of the second stage problem. Note that the first-stage optimization does not generate a fixed cycle length, but the cycle length is bounded by the minimum and maximum green times of each phase. In the second stage, the trajectory of each leading vehicle is controlled to minimize fuel consumption and emission in Eq. (3).

$$\min_{u_p} E(g_p, u_p) \tag{3}$$

$$\text{s.t. } \boldsymbol{h}(g_p, u_p) \leq 0 \tag{4}$$

As shown in Figure 5.9, the objective of the trajectory control is to make the leading vehicle arrive at the intersection (distance L) at time point t_f, which is the beginning of green, with the optimal speed v_{tf} (Figure 5.9 (b)). v_{tf} is defined as the speed at which the flow rate reaches the saturation flow rate q_s as shown in Figure 5.9. For simplicity, some assumptions are made. All vehicles are homogenous, which have the same size and vehicle dynamics (e.g. acceleration, desired speed). All following vehicles obey certain car following rules, based on which following vehicles with larger gaps will try to catch up with their leading vehicles with safety constraints. Lane changing and overtaking behaviors are prohibited. Therefore, a compact platoon can be generated naturally without controlling the trajectories of all vehicles.



(a) (b)

Figure 5.9: Leading Vehicle Trajectory Control

Constraint (2) mainly includes signal timing parameters (e.g. minimum and maximum green time) and car-following rules. Constraint (4) presents vehicle dynamics limits (e.g. maximum acceleration), travel time and travel distance.

Leading vehicle Trajectory Control

The trajectories of platoon leading vehicles are controlled to arrive at the intersection at the beginning of green with an optimal speed. An optimal control model is formulated, with the objective to minimize acceleration fluctuation:

$$\min_{u(t)} J = \int_{t_0}^{t_f} |u(t)|\, dt \tag{5}$$

Vehicle position $x(t)$ and speed $v(t)$ at time t are the state variables. Vehicle acceleration rate $u(t)$ is the control variable and is bounded by maximum deceleration $-a^L$ and acceleration rates a^U. The relationship between the state variables and the control variable defines the vehicle dynamics. Initial and final states are defined based on vehicle's current speed, distance to the intersection from the current position, arrival time, and arrival speed.

The purpose of using this objective function is to derive analytical solutions by the *Pontryagin's minimum principle* (PMP) (Sethi and Thompson, 2000), which greatly reduces the computational time. Theoretical proof shows that the optimal trajectory consists of no more than three segments. As an example, Figure 5.10 shows a general optimal trajectory in which the switch time t_1 and t_2 can be obtained uniquely by solving Eqs. (6) and (7) with the constraint of $t_1 < t_2$. The detailed model formulations can be found in (Feng et al., 2018).

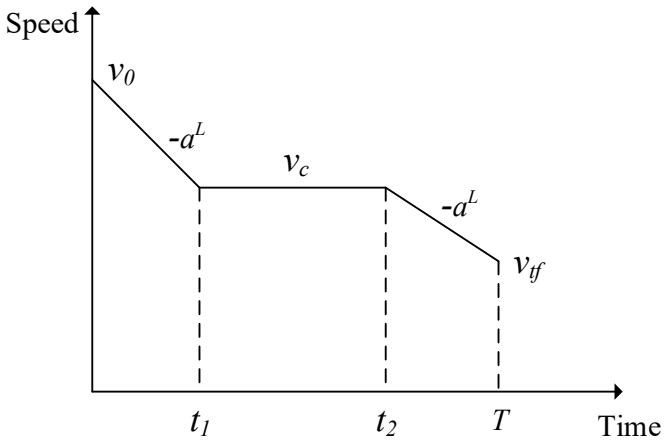


Figure 5.10 A General Optimal Trajectory

$$\frac{v_0 + v_c}{2} t_1 + v_c(t_2 - t_1) + \frac{v_{tf} + v_c}{2}(t_f - t_2) = L \tag{6}$$

$$v_c = v_0 - a^L t_1 = v_{tf} + a^L(t_f - t_2) \tag{7}$$

To justify the simplification, EPA's MOVES emission model (EPA, 2002) is applied as the objective function for comparison as described in detail in the next section.

Benchmark Model for Objective Function Justification

To justify the simplified objective function in Eq. (5), the multi-scale motor vehicle & equipment emission system (MOVES) model (EPA, 2002) of U.S. Environmental Protection Agency (EPA) is applied as the benchmark model for comparison. This model estimates vehicle specific power (VSP) with the input from vehicle speed and acceleration values. Then VSP modes are identified through a look-up table. Furthermore, according to different types of vehicles, engine sizes, and mileages, vehicle emission is located in the emission table (refer to the EPA report (EPA, 2002) for more details).

With the MOVES emission model as the objective function, the analytical solution is not available. Therefore, an approximation model (He et al., 2015) which transforms the optimal control problem to a non-linear programming problem with much fewer decision variables is constructed for platoon leading vehicle trajectory control. The approximation model also divides the vehicle trajectory into three segments with constant acceleration or deceleration rates in each segment. The middle segment is for vehicle cruising where the vehicle speed is kept as a constant. Therefore, the decision variables in this model are reduced to four: two acceleration/deceleration rates a_1, a_2, and two switch time points t_1, t_2. Detailed model formulations can be found in (Feng et al., 2018). Although the formulation is simple and the number of decision variables is small, this problem is non-linear which increases the computational burden, and yet the quality of the solutions cannot be guaranteed, which is demonstrated in the numerical examples.

Platoon Identification

In order to identify the leading vehicle of each platoon to apply the optimal control model, a platoon identification algorithm is developed to separate platoons for different cycles within the DSRC communication range, as shown in Figure 5.11. Platoons are identified one by one from the stop-bar. The number of identified platoons is related to the DSRC communication range as well as the number of cycles planned in DP. Generally, the platoon number should be less than or equal to the cycle

number planned in DP. In reality, due to the limited range of DSRC communication (e.g. 300 m), usually only 1 or at most 2 platoons can be identified for each phase with a reasonable cycle length. The following illustrations are based on only one cycle planned in DP. The analysis is similar if more cycles are planned.

The platoon size for phase p is limited by several factors:

1. The duration of the green time generated by DP in signal optimization. The maximum number of vehicles can be calculated as $floor(g_p/h_s)$, where g_p is the remaining green time of phase p, h_s is the saturation headway and the *floor* function means rounding down to the next integer.
2. Vehicle location. If a vehicle is too far away from the stop-bar, then it may not be able to catch the platoon. The furthest distance d_{max} that a vehicle can be included in the first platoon is calculated as $d_{max} = (r_p+g_p)\times v^f$, where r_p is the remaining red time of phase p and v^f is the free flow speed. If phase p is the current phase, then $r_p=0$.
3. DSRC range d_{DSRC}. If a vehicle is outside the DSRC range, then it will not be included in the platoon identification algorithm.

In summary, if a vehicle's sequence in the approaching vehicles is less than or equal to $floor(g_p/h_s)$ and its distance to the intersection is small than $min(d_{max}, d_{DSRC})$, this vehicle can be included in the first platoon. Otherwise, it should be included in other platoons and pass the intersection without stops at the stop-bar in later cycles. Therefore, no queues will be generated at the stop-bar.

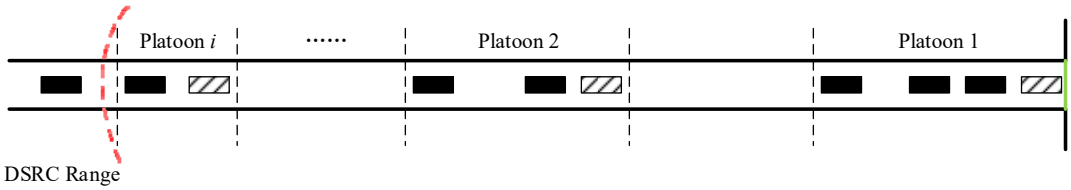


Figure 5.11: Platoon Identification

Car-Following Model

In order to model the behaviors of following vehicles in a platoon, the Next Generation Simulation (NGSIM) car-following model (Yeo et al., 2008) is adopted to update their trajectories. This car-following model is based on Newell's linear car-following (Newell, 2002) model with additional safety constraints to avoid collisions (Gipps, 1981). The model also considers the vehicle performance limits such as maximum acceleration and deceleration rates. The detailed model formulations can be found in (Yeo et al., 2008).

Signal Optimization

The signal optimization is formulated as a DP problem which considers each phase as a stage in DP (Sen and Head, 1997). A forward recursion is used to calculate the performance measures and record the optimal value function. A backward recursion is used to retrieve the optimal solution. The detailed model formulations can be found in (Feng et al., 2015).

Rolling Horizon Scheme

The proposed joint control algorithm is implemented in MATLAB. The flow chart of the optimization process is shown in Figure 5.12. Time is discretized into 1s steps and signal status and vehicle trajectories are updated every time step. A rolling horizon scheme is adopted in which the optimization process is repeated every 5 seconds to include recent vehicle arrivals. The planning horizon for signal optimization is two cycles, in which two cycles of signal timing are generated by DP. The generated signal timing will be executed in the next rolling horizon (5 seconds). It also serves as the input of the platoon identification algorithm. Then the optimal trajectories of the platoon leading vehicles are

solved analytically using the optimal control model. Similarly, the platoon leading vehicles will follow the optimal trajectory during the next rolling horizon. Following vehicles update their trajectories according to the car-following model. The vehicle trajectories are used to estimate the performance function in signal optimization.

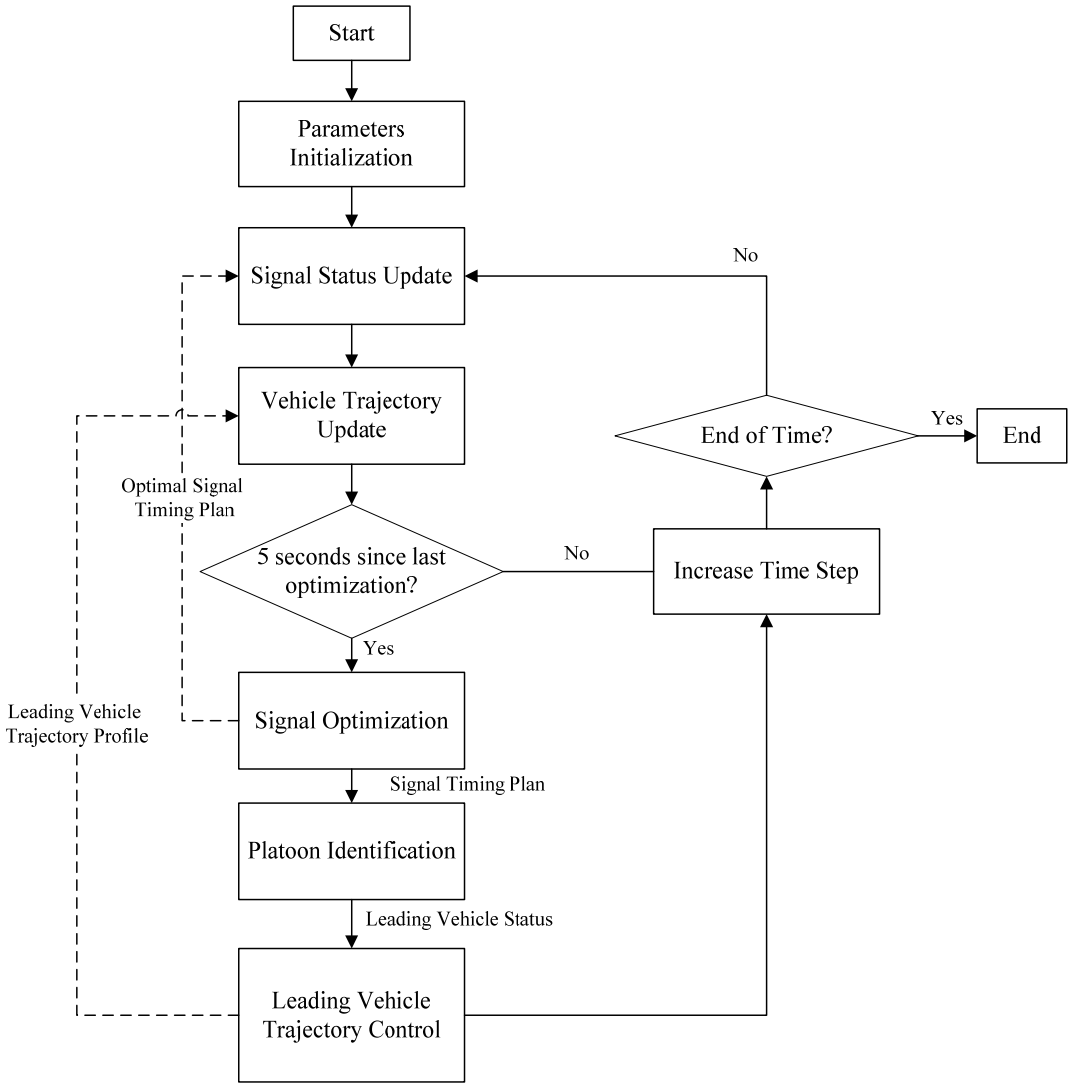


Figure 5.12: Rolling Horizon Scheme

Numerical Examples

Simulation Setup

In the following numerical examples, a hypothetical intersection of two single-lane approaches is used. Two signal phases are applied and no turning movements and lane changing behaviors are considered. The DSRC range is 300 m from the center of the intersection which provides reliable communication (Emmelmann et al., 2010). All vehicles in the communication range are controllable, although only a few vehicles are controlled.

The default parameters for the car-following model are set as follows: τ_n=2 s, $l_{n-1} + g_n^{jam}$=6 m (consider uniform vehicle length), a_n^U=2 m/s^2, a_n^L=-2 m/s^2, v_n^f=14 m/s (~50 km/h).

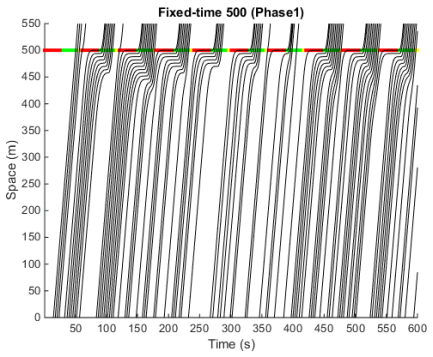
In signal optimization, the two phases of a cycle are identical. The corresponding default parameters are set as follows: the minimum green time g_{min}=10, the maximum green time g_{max}=26, and the

transition interval g_{tran}=4 which includes yellow interval and all-red clearance time. Therefore, X_j^{min}= g_{min}+ g_{tran} and X_j^{max}= g_{max}+ g_{tran}. The maximum acceleration and deceleration rates for vehicle trajectory control are set the same as those in the car-following model, and the optimal speed v_{tf}=10 m/s.

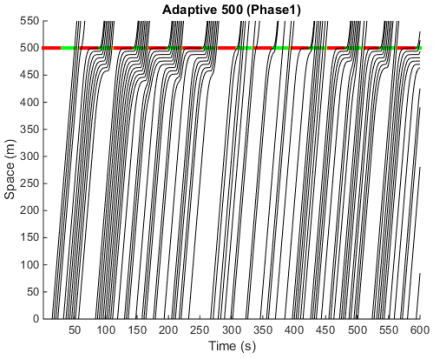
The optimal control formulation for leading vehicle trajectory control may not always have solutions. Based on the parameters above, if the green time generated by DP is smaller or equal to 5 seconds, the signal optimization will not be executed until the beginning of the next phase to prevent modifying the leading vehicle trajectory when it is too close to the intersection.

Simulation Results and Discussion

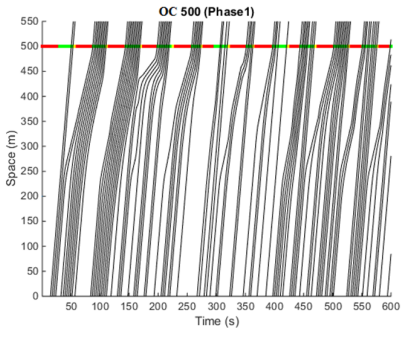
Vehicle arrival conforms to the Poisson distribution. Three different traffic demand levels are tested. The demands in the two approaches are set to be the same. The three levels are 500 veh/h/lane, 650 veh/h/lane and 800 veh/h/lane (i.e., medium, high and saturated traffic conditions). The corresponding v/c ratios are 0.64, 0.83, and 0.97. v/c ratio is calculated based on an 1800 veh/h/lane saturation flow rate and the effective green time is equal to the actual green time. Four different scenarios are simulated: Fixed, Adaptive, OC and NLP. In the "Fixed" scenario, vehicle trajectories are not controlled, and the signal timing is fixed. Each phase has 26s of green time, 4s of transition time and 30s of red time. In the "Adaptive" scenario, vehicle trajectories are not controlled, but the signal timing is optimized using DP. In the "OC" scenario, vehicle trajectories are controlled using the optimal control model with simplified objective function and the signal timing is optimized using DP. In the "NLP" scenario, vehicle trajectories are controlled using the Non-linear programming (NLP) approximation model with MOVES model as the objective function and the signal timing is optimized using DP. The total simulation time for each scenario is 1000s. Figure 5.13 shows the comparison of vehicle trajectories under the four scenarios with medium demand level.



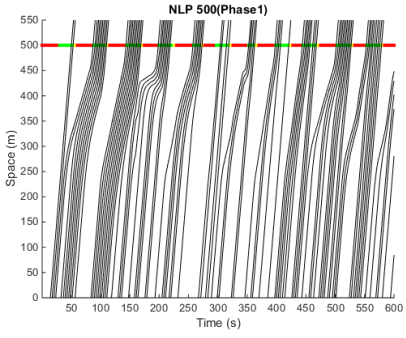
(a) Fixed



(b) Adaptive



(c) OC



(d) NLP

Figure 5.13: Vehicle Trajectory Comparison under Four Scenarios

By comparing the scenarios "Fixed" and "Adaptive", it can be seen that in some cycles of the "Fixed" scenario, a portion of green time is wasted. The "Adaptive" scenario generates the timing plan adaptively based on vehicle arrivals so that green time will be utilized more efficiently. Without controlling vehicle trajectories, vehicles stop at the stop-bar for the red signals in both "Fixed" and "Adaptive" scenarios. With vehicle trajectories controlled, leading vehicles of each platoon in the "OC" and "NLP" scenarios slowdown in the middle of the road segment to avoid stops at the stop-bar. They are controlled to arrive at the intersection at the beginning of the green most of the times to improve the green time utilization. Following vehicles obey the car-following model to catch up with the preceding vehicles so that compact platoons are generated. The figure also shows that scenarios "OC" and "NLP" generate similar vehicle trajectories. In both scenarios, at the end of the third cycle (around 150s), there is a sudden deceleration of the leading vehicle trajectory. That's caused by the change of the signal timing due to adaptive signal control. In this case, the green time generated by the adaptive control algorithm is 1 second shorter than the previous rolling horizon. As a result, the last vehicle of the previous platoon cannot pass the intersection and it becomes the leading vehicle of a new platoon. It is then controlled to arrive at the intersection in the next cycle. The leading vehicle of the sixth cycle (around 270s) cannot arrive at the intersection at the beginning of the green so that it is not controlled and it travels at free-flow speed to the intersection.

Table 5.2 shows the comparison of total vehicle delay, CO_2 emission and execution time of the four scenarios under different traffic demand levels. Results of all scenarios are the average of 5 different random seeds. CO_2 emission is calculated by the MOVES emission model based on the vehicle trajectories generated in each scenario. Note that the "NLP" scenario incorporates MOVES model to the objective function while other three scenarios just use MOVES model for evaluation. Vehicle category 11 in MOVES model (odometer<50,000 miles and engine size <3.5 liters) is used. The execution time to run the 1000s simulation is recorded.

Table 5. 2: Comparison of Vehicle Delay, CO_2 Emission and Execution Time

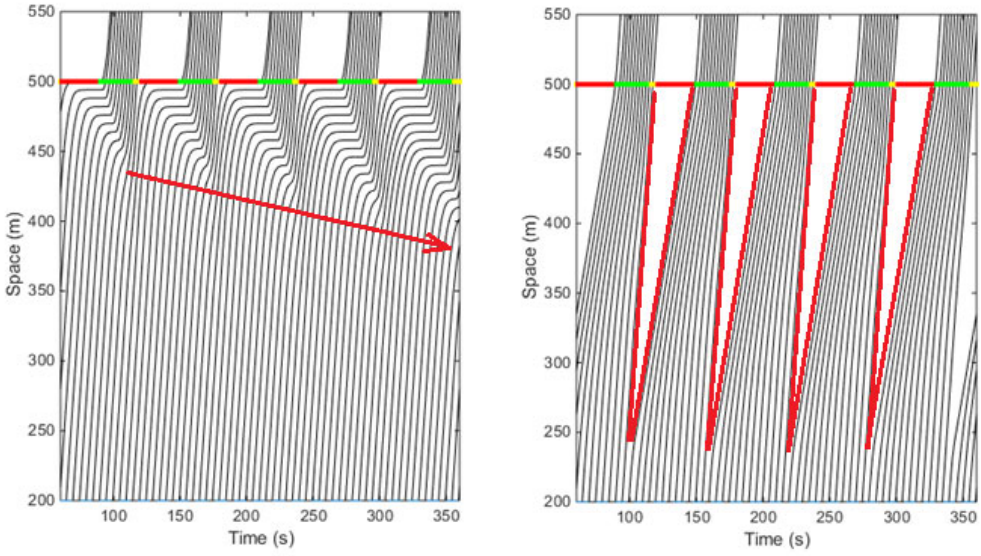
Scenario	Demand Level: 500 veh/h/lane(v/c=0.64)*				
	Delay (s)	**%**	**CO_2 Emission (kg)**	**%**	**Execution time (s)**
Fixed	4400.6	N/A	51.4	N/A	0.8
Adaptive	3746.4	-14.9	47.5	-7.5	6.1
OC	3752.2	-14.7	47.2	-8.2	1.4
NLP	3851.4	-12.5	47.9	-6.7	657.8
	Demand Level: 650 veh/h/lane (v/c=0.83)				
Fixed	15761.0	N/A	113.5	N/A	1.1
Adaptive	14940.6	-5.2	110.1	-2.9	6.3
OC	11981.6	-24.0	97.9	-13.8	1.6
NLP	12056.0	-23.5	99.1	-12.6	886.6

	Demand Level: 800 veh/h/lane (v/c=0.97)				
Fixed	31729.2	N/A	159.6	N/A	1.7
Adaptive	31549.0	-0.6	157.0	-1.6	6.6
OC	28381.4	-10.6	150.1	-6.0	2.3
NLP	27635.8	-12.9	150.4	-5.8	1529.9

*v/c ratio is calculated assuming an 1800 veh/h/lane saturation flow rate and effective green time is equal to actual green time.

Several observations can be made from the results:

1. Without vehicle trajectory control, adaptive control outperforms fixed-time control in terms of both vehicle delay and CO_2 emission. The benefit decreases as traffic demand increases. That's because, under higher demand levels, adaptive control tends to assign maximum green time to each phase to serve more demand which essentially turns to be fixed-time control. With vehicle trajectory control, under saturated demand level, "OC" and "NLP" can still reduce about 10% vehicle delay compared to "Fixed" and "Adaptive". Because the trajectory control eliminates the startup lost time and increases the capacity of the intersection. To better illustrate the benefit of capacity increase, we design a special case with uniform vehicle arrival (800 veh/h/lane) and fixed-time signal as shown in Figure 5.13. It can be seen from Figure 5.13 (a) that, without trajectory control, the intersection is oversaturated and the queue is propagating over cycles Figure 5.13 (b) shows the vehicle trajectories under the same demand level and the same signal plan. The gaps between each trajectory block remain the same over cycles, which suggests no oversaturation in this case. It is well known that vehicle delay increases dramatically under oversaturated traffic conditions, which can be avoided because of the increased capacity.



(a) (b)

Figure 5.13 Capacity Increase with Trajectory Control

2. Both vehicle delay and emission are reduced in the “OC” and “NLP” scenarios compared to “Fixed” and “Adaptive” scenarios by as much as 24.0% and 13.8%, respectively. More benefits are shown under high demand level. Compared to the medium and saturated demand levels, both the signal optimization and vehicle trajectory control have more flexibility in terms of green time allocation and intersection capacity utilization.
3. The vehicle delays in “OC” and “NLP” scenarios are similar. But “OC” scenario generates lower emission than “NLP” scenario in all cases. In both vehicle trajectory optimization problems, vehicle delay is formulated as a constraint, because the arrival time at the intersection is fixed through trajectory control As long as a feasible solution can be found, both problems generate similar vehicle delays. However, emissions are formulated as the objective function, whose value depends on the quality of the solution. The approximation model is a NLP problem and no global optimality is guaranteed. In spite of the simplified objective function in the optimal control formulation, the analytical solutions still outperform those generated using the emission model as the objective function.
4. The execution time differs among the four scenarios. The “Fixed”, “Adaptive”, and “OC” scenarios have similar execution times while “NLP” scenario requires significantly longer time. In the “NLP” scenario, a long execution time is observed due to the difficulty in dealing with nonlinearity. On the contrary, the analytical solution from the optimal control formulation reduces computational time notably.

The results have validated the use of the simplified objective function instead of the exact but complex emission model in terms of both computational time and solution performance.

5.2.2 Integrated Optimization

This study presents a mixed integer-linear programming (MILP) model to optimize traffic signals and vehicle trajectories at isolated urban intersections in a unified framework. Phase sequences, green start and duration of each phase, and cycle lengths are optimized together with vehicle lane-changing behaviors and vehicle arrival times in the MILP model. Vehicles are guaranteed to pass through an intersection at desired speeds and avoid stops at stop bars. A new planning horizon strategy is applied to conduct the optimization. Platoons in each lane are identified based on the optimization results. Exact vehicle trajectories are then generated by optimal control models and car-following models. The trajectory of each platoon leading vehicle is optimized by an optimal control model with the objective to minimize fuel consumption and emission. Lower and upper bounds of arrival time constraints are imposed to the MILP model in order to generate feasible trajectories. Clearance time is considered to eliminate conflictions between incompatible vehicle movements within an intersection area. The proposed MILP model is similar with slot-based “signal free” models if the minimum green time constraint is removed. However, the proposed model optimizes vehicle trajectories at the upstream of an intersection instead of managing trajectories within an intersection area.

Problem Description

For a typical intersection, there are three vehicle movements (i.e., left-tuning, through, and right-turning) in each arm. Each movement has a different desired speed to pass through the intersection for safety concerns. As shown in Figure 5.14, in each arm, the approach lane index is incremented from the left most lane. Each approach lane is dedicated to one vehicle movement. The distance between a vehicle and the stop bar at time t is denoted as $x^{\omega}(t)$. L_i is the control zone in arm i in which vehicle trajectories can be optimized. L_i^p is the no-changing zone in which vehicles keep their previously optimized trajectories. That is, only the trajectories of vehicles that are outside the no-changing zone will be updated over time. The no-changing zone is designed to reduce computational burden but at the cost of optimality.

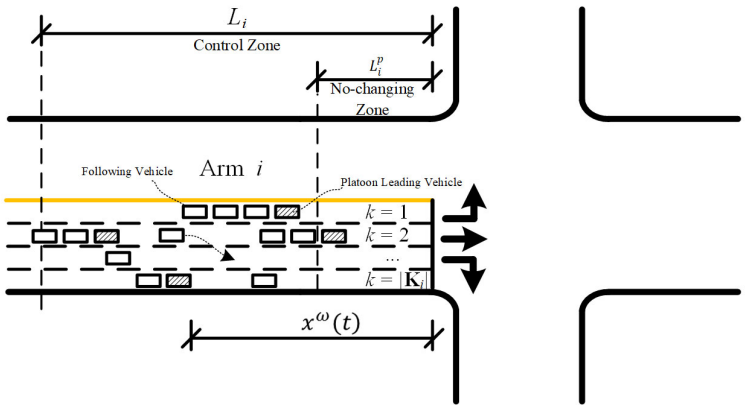


Figure 5.14 An intersection with four arms.

The task is to integrate vehicle trajectory planning into traffic signal optimization in a unified framework to minimize vehicle delays. The model framework is shown in Figure 5.15. The trajectory planning serves for two purposes: 1) build the relationships between vehicle arrival times in the same lane, which helps build the arrival time constraints (e.g., upper and lower bounds) in the MILP model; 2) generate trajectories for vehicles in the same lane given their arrival times, which are supposed to be optimized in the MILP model. Note that no exact vehicle trajectories or platooning are needed in building the constraints of arrival times in the MILP model. The outputs of the optimization model include signals, vehicle lane choices, and vehicle arrival times. Platoons, i.e., the vehicles that pass the stop bar in the same lane in the same cycle, are identified based on the optimization results. However, only vehicle lane choices and vehicle arrival times are needed in planning exact trajectories (i.e., acceleration, speed, and location profiles) for vehicles in each lane. Platooning is not necessarily needed when generating exact vehicle trajectories.

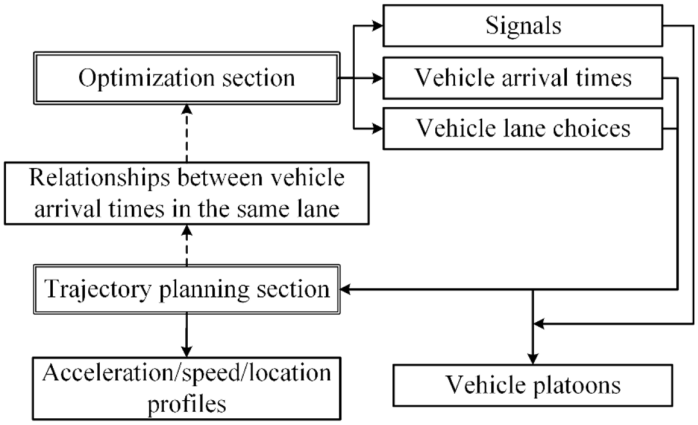


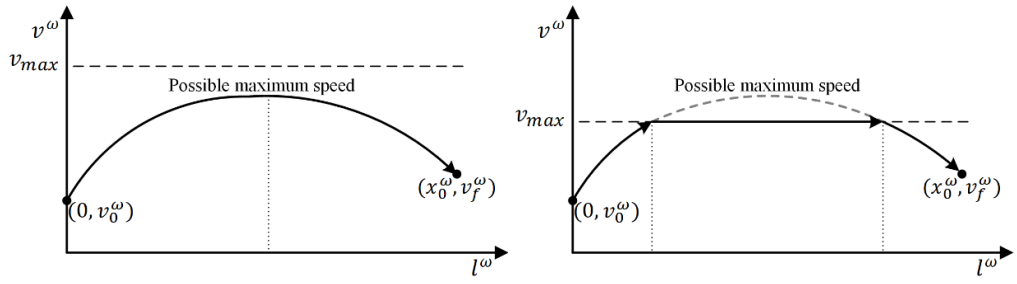
Figure 5.15 Model framework.

Model Formulation

Vehicle Trajectory Planning

The optimal control model introduced in Section 5.2.1 is applied to generate optimal trajectories for platoon leading vehicles with the objective of minimizing fuel consumption/emission given arrival times. The Newell's car-following models are used to capture the trajectories of following vehicles. In this way, the identification of vehicle trajectories is equivalent to determining the arrival times. To guarantee the feasibility of the optimal control models, this section also helps build the constraints of vehicle arrival times for the MILP model. The additional constraints address the problem that the speed of an optimized trajectory may exceed the post speed limit. Two cases are considered and shown in

Figure 5.16. In the first case, the possible maximum speed never exceeds the speed limit. The vehicle can exactly follow the optimized trajectory. In the second case, the possible maximum speed may exceed the speed limit, then the original planned speed is replaced by the speed limit during the time interval that the situation happens. Detailed derivation of the optimal trajectory can be found in (Yu et al, 2018).



(a) (b)

Figure 5.16 Illustration of two cases: (a) Case 1, and (b) Case 2.

Optimization Model Formulation

This section presents the optimization model for vehicle arrival times and traffic signal parameters (i.e., phase sequences, green starts, green durations, and cycle lengths) considering lane-changing behaviors. The objective is vehicle delay minimization. Vehicle trajectory planning is integrated by optimizing vehicle arrival times and lane choices, which determines vehicle delays. The optimization is conducted based on a planning horizon strategy. Basic assumptions, vehicle trajectory constraints, signal constraints, and the objective function are introduced sequentially.

The following assumptions are made to simplify the problem:

a) Vehicles are homogeneous. The desired speed when passing the intersection is only related to its movement direction.

b) All vehicles are controllable.

c) Vehicles and the signal controller can communicate information in real time within the control zone.

d) Vehicles are in permitted lanes when entering the control zone in each arm. The permitted lanes are the dedicated lanes for a certain vehicle movement.

e) Lane changing behaviors are assumed to be completed instantly.

The primary objective of the proposed optimization model is the minimization of vehicle travel delays. The travel delay of each vehicle is defined as the difference between the actual travel time and the free flow travel time. It is noticed that there may be multiple solutions that have the same total vehicle delay but with different cycle lengths. Since we have no constraints of maximum cycle lengths or maximum phase green times to make the model more flexible, a secondary objective of cycle length minimization is added. It increases the frequency of switching right of way and potentially decreases the delays of the incoming vehicles in the future. The proposed model is a hierarchical multi-objective optimization model. Total vehicle delay is first minimized and then cycle lengths are minimized for the solutions with the same minimum delay. The most common approach to such a model is the weighted sum method (Marler and Arora, 2004). As a result, the objective function is formulated as

$$\min_{\boldsymbol{V}=(\overline{\boldsymbol{T}},\boldsymbol{S})} \alpha_1 \sum_{i\in\mathbf{I}} \sum_{\omega\in\boldsymbol{\Omega}_i} \left(t_f^{\omega} - t_e^{\omega} - \frac{L_i}{v_{max}}\right) + \alpha_2 \sum_{n=1}^{N} C^n \tag{7}$$

where $\overline{\boldsymbol{T}}$ is a subset of the trajectory variable set $\boldsymbol{T}$ (i.e., $\overline{\boldsymbol{T}} = \boldsymbol{T} - \{x^{\omega}(t), l^{\omega}(t), v^{\omega}(t), a^{\omega}(t)\}$); α_1 and α_2 are weighting parameters and $\alpha_1 \gg \alpha_2 > 0$. t_f^{ω} is the optimized arrival time of vehicle ω; t_e^{ω} is the generation time of vehicle ω; L_i is the control zone in arm i in which vehicle trajectories are to be controlled; v_{max} is the maximum allowed vehicle speed; and C is the cycle length.

Constraints include vehicle trajectory constrains (permitted occupied lanes, possible target lanes, lane changing behaviors, gap acceptance conditions for lane changing, lower and upper bounds of vehicle arrival times, and no-changing zones), and signal constrains (lane signal settings, green start time, duration of green, green end time, cycle length, phase sequence, clearance time, and vehicle arrival times at stop bars). Detailed optimization model formulation can be found in (Yu et al, 2018).

Planning Horizon Procedure

Previous models use time (in seconds) as the length of the planning horizon with a fixed number of cycles (Feng et al., 2015), which is different in this study. The cycle number *N* in the planning horizon, as shown in Figure 5.17, depends on the number of vehicles considered in the optimization. Note that cycle lengths are optimized over time and, therefore, the total time of the planning horizon may vary over time as well. Based on the constraints in terms of vehicle arrival times at stop bars, *N* needs to be large enough so that all vehicles are planned to pass through the intersection in the *N* cycles. A smaller *N* may render the optimization model infeasible while a larger *N* increases computational burden. As a result, we choose the smallest *N* that makes the model feasible. Note that the choice of *N* has no impacts on solution optimality under the condition that the MILP model is feasible. The planning horizon procedure is shown in Figure 5.18, which follows:

Step 1: Initialize horizon start time $t_s = 0$ and cycle number $N = 1$ at initial time $t_0 = 0$.

Step 2: Collect information of vehicles in the control zone at time t_0.

Step 3: Solve the special case of the MILP model **P3** with $\alpha_1 = 1$ and $\alpha_2 = 0$.

Step 4: If the model is infeasible, then update the cycle number $N = N + 1$ and go to **Step 3**. Otherwise, go to the next step.

Step 5: Select α_1 and α_2 so that α_1/α_2 is large enough.

Step 6: Solve the MILP model **P3**.

Step 7: Update the signals and the planned vehicle trajectories in the control zone according to the optimization results.

Step 8: Record the optimized length C^1 of the first cycle.

Step 9: Update time $t_0 = t_0 + \Delta t$, where Δt is the time step.

Step 10: If t_0 reaches the final simulation time, then end the process. Otherwise, go to the next step.

Step 11: If $t_0 \geq t_s + C^1$, then update $t_s = t_s + C^1$. Go to **Step 2**.

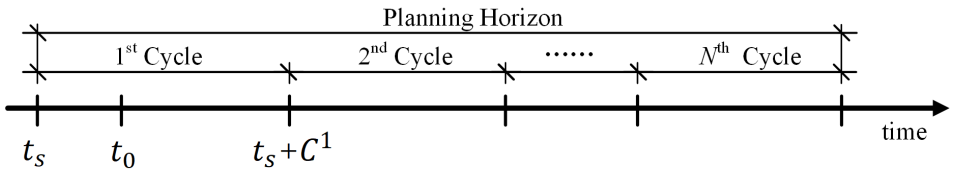


Figure 5.17 Illustration of planning horizon.

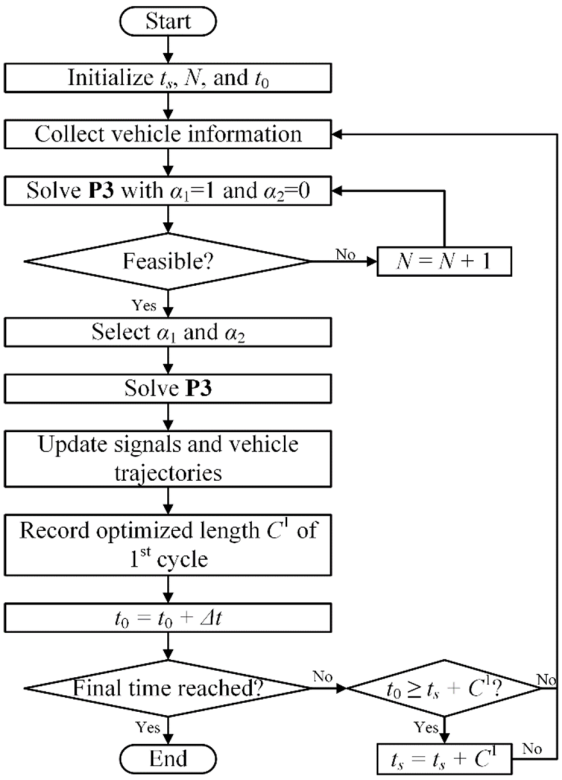


Figure 5.18 Planning horizon procedure.

Numerical Examples

Experimental Data

To evaluate the proposed models, a typical four-arm intersection with all directions of movements is applied. The lane markings of the intersection are shown in Figure 5.18. Right-turning vehicles are not controlled by traffic signals but they are controlled to arrive at the intersection at a desired speed. The length L_i of the control zone in each arm is 300 m which is the reliable communication range of Dedicated Short-Range Communications (DSRC) (Emmelmann et al., 2010). The length L_i^p of the no-changing zone in each arm is 50 m.

The basic traffic demand and the volume/capacity (v/c) ratios are shown in Table 5.3. The v/c ratios are calculated with the assumed green duration of 26 s for each phase and a cycle length of 120 s. The saturation flow in each lane is determined by the time headway. Vehicles are generated according to Poisson distribution, which is a common practice for traffic control at isolated intersections (Jiang et al., 2017; Li et al., 2014). Speed limit v_{max} is 15 m/s. The desired speeds v_f^ω of left-turning, though, and right-turning vehicles passing through the intersection are 10 m/s, 13 m/s, and 8 m/s, respectively. Vehicles enter the control zone at the speed of 13 m/s instead of the speed

limit. The time displacement τ^{ω} and space displacement d^{ω} in the car-following model are 0.9 s (assuming quick reaction of CAVs) and 6 m. The minimum time interval between two lane changing behaviors of a vehicle is 5 s. The absolute values of the maximum comfortable acceleration and deceleration rates (a_U and a_L) are 2 m/s^2 and 4 m/s^2. The clearance time between incompatible traffic flows (e.g., the traffic flow from arm 1 to arm 2 and the traffic flow from arm 2 to arm 3) is 4 s. The minimum green time $g_{min}^{i,j}$ is 6 s. The tolerance Δd of solution quality degradation is 3 s. The weighting parameters α_1 and α_2 in the objective function Eq. (7) are 300 and 1 in each optimization. To investigate the environmental impacts, CO_2 emission model in Frey et al. (2002) is employed.

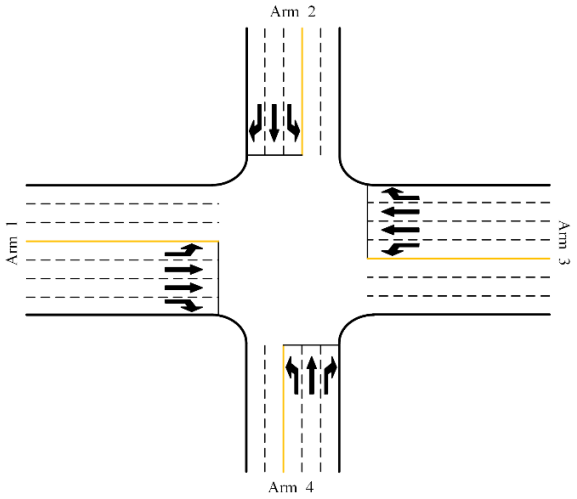


Figure 5.18 Lane markings of a four-arm intersection.

Table 5.3 Basic traffic demand and volume/capacity ratios

Traffic demand in pcu/h (v/c)	To Arm			
From Arm	1	2	3	4
1	-	200 (0.38)	400 (0.35)	100 (0.21)
2	150 (0.32)	-	150 (0.29)	200 (0.35)
3	380 (0.33)	150 (0.32)	-	180 (0.35)
4	100 (0.19)	200 (0.35)	100 (0.21)	-

The optimization models are written in C# and solved using Gurobi 7.5.1 (Gurobi Optimization Inc., 2017). All the experiments are performed in a desktop computer with an Intel 3.6 GHz CPU and 16 GB memory. An upper limit of 1.5 s is set for real-time application. A sub-optimal solution produced by the solver will be accepted if the solving time exceeds the time limit.

Results and Discussion

Vehicle-actuated control is applied in the simulation as the benchmark for comparison with the proposed control method, denoted as CAV-based control. In actuated control, the maximum green durations for arm 1 and arm 3 are 30 s, and the maximum green duration for arm 2 and arm 4 are 20 s. The minimum green duration is 4 s which is the optimal value by trial and error. The unit extension time is 2 s. The time of each simulation scenario is 1200 s and the simulation time step is 1 s. Average

values of throughput, vehicle delays, and CO_2 emissions of ten different random seeds are recorded and shown in Table 5.4–Table 5.6. Five levels of traffic demand are tested, which are the product of the basic demand and a demand factor. The demand with the factors from 0.6 to 2.0 is under-saturated and the demand with factors 3.0 and 4.0 are over-saturated. The under-saturated and over-saturated traffic condition is observed with actuated control.

Table 5.4 Throughput

Demand factor	Throughput (veh)		
	Actuated Control	CAV-based Control	Increase (%)
0.6	289.60	291.20	0.55
1.0	446.00	449.20	0.72
2.0	686.00	691.50	0.80
3.0	764.60	824.80	7.87
4.0	769.70	922.10	19.80

Table 5.5 Average delay

Demand factor	Average Delay (s/veh)		
	Actuated Control	CAV-based Control	Decrease (%)
0.6	14.65	8.62	41.16
1.0	15.84	11.33	28.47
2.0	18.32	13.59	25.82
3.0	61.71	15.40	75.04
4.0	98.28	16.34	83.37

Table 5.6 Average CO_2 emissions

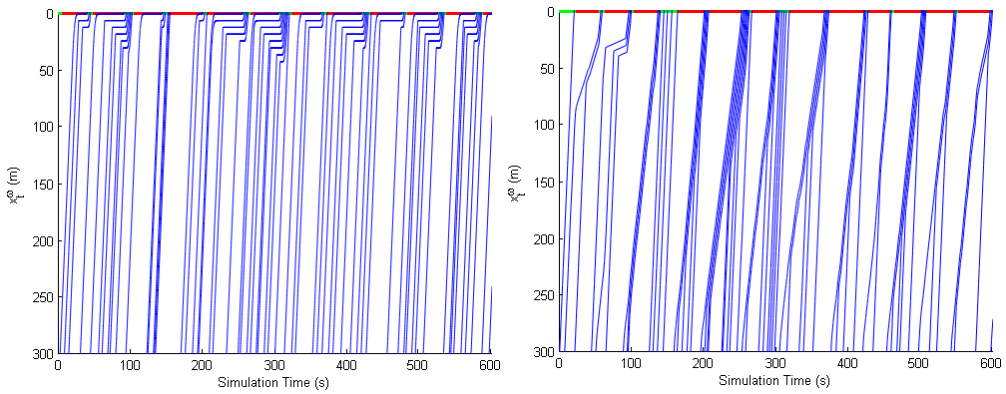
Demand factor	Average CO_2 Emissions (g/veh)		
	Actuated Control	CAV-based Control	Decrease (%)
0.6	123.96	114.73	7.45
1.0	126.19	123.51	2.12
2.0	134.98	132.38	1.93
3.0	212.65	133.08	37.42

4.0	267.07	135.39	49.31

Table 5.4 shows that CAV-based control improves intersection capacity. The throughput increase under CAV-based control is insignificant with under-saturated traffic demand. This is consistent with our intuition because demand is below intersection capacity under both control methods. When the demand factor further increases to 3.0 and 4.0, intersection capacity is reached under actuated control because the throughput almost remains the same. In contrast, the throughput under CAV-based control keeps increasing noticeably. This indicates that intersection capacity under CAV-based control is higher compared with actuated control.

Table 5.5 shows the significant decrease of vehicle delays when CAV-based control is applied, which can reach ~40% under low traffic demand and ~80% under high demand. The benefits are mainly due to improved intersection capacity as well as the more efficient use of green time at the intersection. Vehicle trajectories are optimized so that all vehicles pass through the intersection at high desired speeds without stops. Thus, no vehicle queues are generated at stop bars, either. As a result, the green start-up lost time is eliminated and more vehicles can pass the intersection during the same green interval compared with actuated control. One interesting observation is that the delay reduction under CAV-based control decreases first and then rises as demand increases. Under under-saturated demand, the benefits are more remarkable with lower demand which indicates strong flow uncertainty. Under over-saturated demand, the delay under actuated control rises more significantly as demand increases. Because the intersection capacity under CAV-based control is much higher than that under actuated control.

Table 5.6 shows the comparison of CO_2 emissions. The results are similar to those in Table 5.5. CAV-based control outperforms actuated control under both under- and over-saturated demand. This is intuitive because the trajectories of platoon leading vehicles are planned with the aim of reducing fuel consumption/emission. Following vehicles also have smoother trajectories since they do not make complete stops at stop bars. But the benefits decrease with increasing traffic volumes under under-saturated demand. The reason is that vehicles are likely to accelerate or decelerate more frequently under CAV-based control when traffic condition varies. As shown in Figure 5.19, vehicles decelerate only once before arriving at the intersection under actuated control while vehicles keep adjusting their speeds according to varying traffic condition under CAV-based control. However, the benefits are significant under over-saturated demand. Because intersection capacity is improved under CAV-based control and thus vehicles experience much less delays. This indicates that vehicles spend much less time traversing the intersection. In this way, CO_2 emissions are greatly reduced.



(a) (b)

Figure 5.19 Trajectories of left-turning vehicles in arm 1 as an example: (a) actuated control, and (b) CAV-based control.

5.2.3 Traffic State Estimation

At signalized intersections, CVs may serve as mobile sensors, providing opportunities of reducing dependencies on conventional vehicle detectors for signal operation. However, most of the existing studies mainly focus on scenarios that penetration rates of CVs reach certain level, e.g., 25%, which may not be feasible in the near future. How to utilize data from a small number of CVs to improve traffic signal operation remains an open question.

In this work, we develop an approach to estimate traffic volume, a key input to many signal optimization algorithms, using GPS trajectory data from CV or navigation devices under low market penetration rates. To estimate traffic volumes, we model vehicle arrivals at signalized intersections as a time-dependent Poisson process, which can account for signal coordination. The estimation problem is formulated as a maximum likelihood problem given multiple observed trajectories from CVs approaching to the intersection. An expectation maximization (EM) procedure is derived to solve the estimation problem.

Methodology

In order to estimate traffic volume, our basic idea is to take advantage of vehicle arrival information in vehicle trajectories. The arrival information can be reflected from the status whether a vehicle stopped or not. An example is shown in Figure 5.20. In the figure, CV1 passed the intersection with a stop and CV2 without a stop. Then, based on CV1's stopping position or departure time, we can calculate number of vehicles queuing in front of it. Based on the trajectory of CV2 without a stop, we know that if vehicle queue existed, the queue would not be long enough to impact CV2. In other words, the upper bound of possible vehicle arrivals between CV1 and CV2 can be calculated based on the trajectory of CV2. By combining these arrival information from vehicle trajectories, volume of overall vehicle arrivals can be estimated.

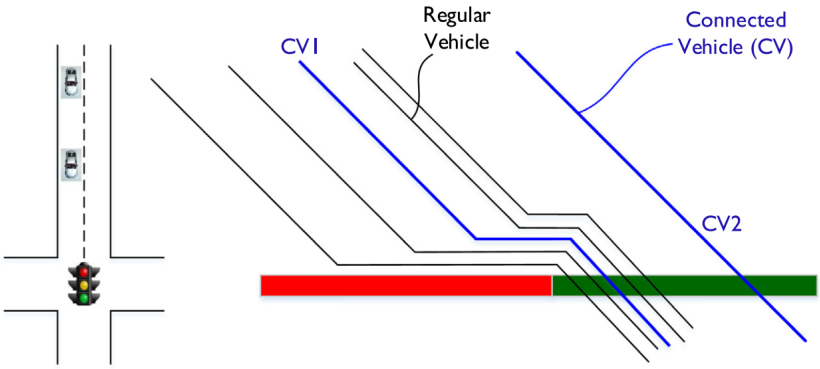


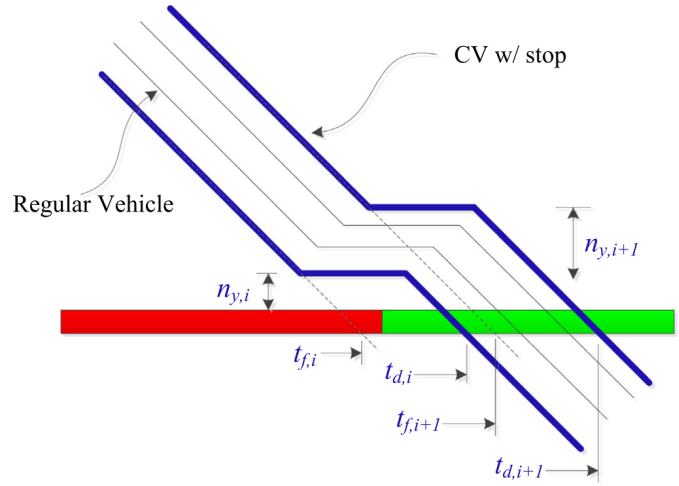
Figure 5.20 Illstration of Vehicle Arrival Information in Trajecotries

The vehicle arrivals are modeled as a time-dependent Poisson process. During a selected Time of Day (TOD) period, we assume that traffic arrivals follow a time-dependent Poisson process with an arrival rate of $\lambda p(t^{(c)})$. Here, $t^{(c)}$ indicates time within a signal cycle, the superscript (c) indicates that the time is measured using a signal clock, k denotes the mean arrival rate, and $p(t^{(c)})$ is the time dependent factor proportional to the arrival rate at $t^{(c)}$, i.e., the fraction of total arrivals at $t^{(c)}$over the entire signal cycle. In traffic engineering literature, Poisson process is a common choice to model traffic arrivals at intersections. The additional assumption that arrival rates are dependent on the time in a signal cycle is to account for impacts from signal coordination. With the signal coordination, traffic departures at the upstream intersection would be grouped as platoons, leading to nonhomogeneous arrivals at subject intersection. The time-dependent Poisson process is used to characterize the non-homogeneous arrivals.

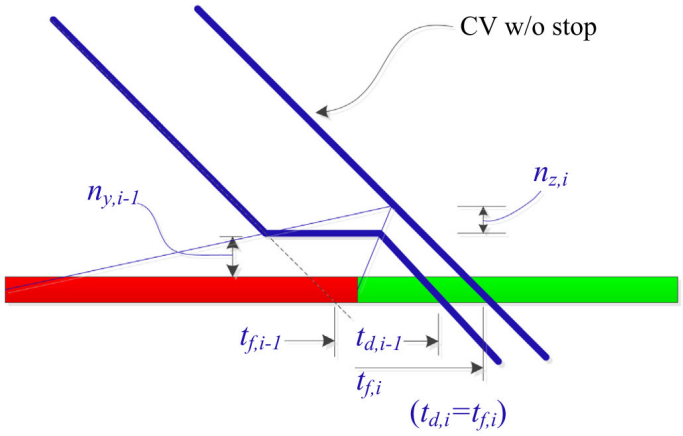
Given the Poisson arrival process, the likelihood function for observing all valid CV trajectories can be formulated by taking advantage of the inter-arrival time and the corresponding number of non-CV arrivals between two consecutive CV trajectories received at RSE. As mentioned earlier, two types of CV trajectories are considered: 1. CV trajectory with a stop at an intersection, and 2. CV trajectory that traverses the intersection without a stop. Between the projected arrival times of two stopped CVs, or between the projected arrival time of one stopped CV and the start of a red signal, the number of non-CV arrivals can be calculated based on the CVs' departure time. If a CV without a stop is observed, then

queues at intersection, if exist, are not long enough to affect the non-stopped CV. Thus, the maximum number of vehicle arrivals before the CV can be calculated. Illustrations of the two types of CVs are shown in Figure 5.21.

Besides these two cases, two other cases of trajectories also exist: 1. stopped CV arriving after a non-stopped CV in the same cycle, and 2. non-stopped CV arriving after another non-stopped CV, also in the same cycle. For the first case, the stop of the CV would not be caused by queues or red signal, but likely by other factors, e.g., mid-block entry of other vehicles. For the second case, after the arrival of a non-stopped CV, we know that the queues must have been cleared and the rest of CVs in the same cycle would travel with free-flow speed. The trajectory therefore does not provide useful information for volume estimation. Accordingly, both cases are considered as invalid or trivial observations, and are not used in the estimation.



(a) CV with a Stop



(b) CV without a Stop

Figure 5.21 Illustrations of Two Different Types of CV Trajectories

Based on the discussion, the likelihood of observing all valid CV trajectories canbe calculated. The arrival rate λ cen be estimated using maximum likelihood estimator (MLE) using the expectation maximization (EM) algorithm. The EM algorithm an iterative procedure to find the MLE mostly suitable when unobserved or partially observed variables exist. The EM algorithm consists of two main steps: the E-step and the M-step. The E-step calculates the conditional expectation of unobserved or partially observed variables based on initialized parameters, and the conditional expectation of the likelihood.

Then, the M-step searches for an optimal update of the parameters through maximizing the likelihood. The two steps are iterated until updates converge. For the details of the EM algorithm, interested readers are referred to Bilmes (1998). In our case, CV trajectories with stop provide direct information of number of arrivals, while trajectories without a stop only provide information of upper bounds of the number of arrivals, i.e., partial information. Considering this, the EM algorithm would be a proper choice for our estimation.

For more details regarding the MLE formulation and the EM algorithm, please refer to (Zheng, et al., 2017).

Case Study

In this case study, we analyzed data from Intersection of Plymouth Rd. & Green Rd., one of the deployed intersections in the SPMD project. CV data used were collected from 04/25/16 to 05/13/16. An illustration of the intersection geometry is shown in Figure 5.22, together with the ring-and-barrier diagram for traffic signal in operation. Here, our investigation focused only on EB through, WB through, as well as SB through and left-turn traffic, corresponding to phase 1, 2 and 4. The NB approach is a single-lane road adjacent to the parking lot of a shopping plaza. At the NB approach, traffic from the driveways and parking lots frequently affected vehicles traveling at the NB approach, resulting in additional queues and vehicle-stops not caused by the traffic signal. Since the stop and queuing information play key roles in our estimation, we exclude the analysis for the NB traffic, considering the noises caused by the traffic from the parking lot.
For each interested approach, trajectories of CVs were first processed as time-space plots with time as the horizontal axis and distance to the stop bar as the vertical axis. The trajectories are shown in Figure 5.23. With the SCOOT adaptive signal system, at this intersection, the cycle length, red and green duration all varied from cycle to cycle. To select a common reference point in a signal cycle, we use the start of green as time 0 in the plot for simplicity. The stop bar position is used as 0 origin along the y-axis. The distance increases upstream along y-axis. That is, vehicles travel from locations of positive distances to negative distances. The CV trajectories were aggregated according to different TOD periods with 1-h intervals across different days, to first calculate time-dependent factors $p(t)$. For different TOD periods, substantially different $p(t)$ were observed with two examples shown in Figure 5.23(b). The differences in $p(t)$ are likely due to differences in both traffic patterns and signal settings in the two different TOD periods. Then, the EM procedure was implemented for the estimation.

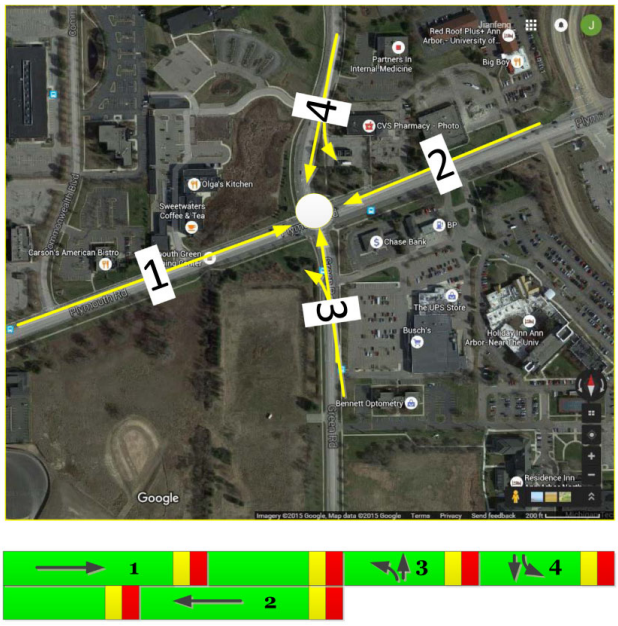
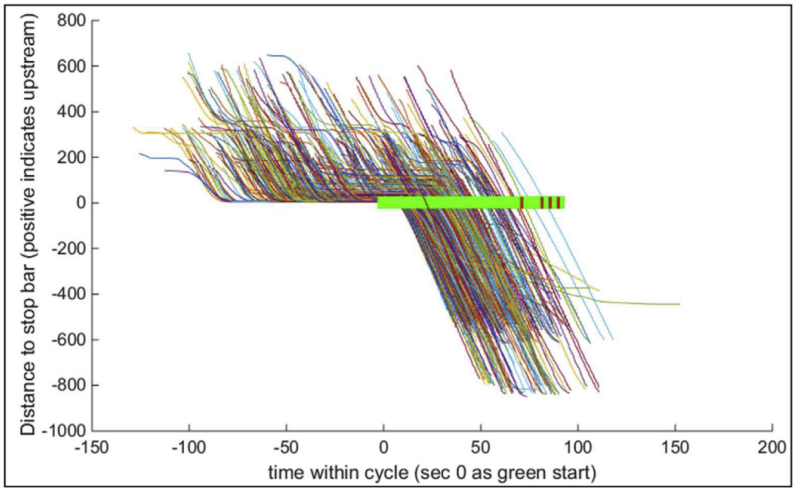
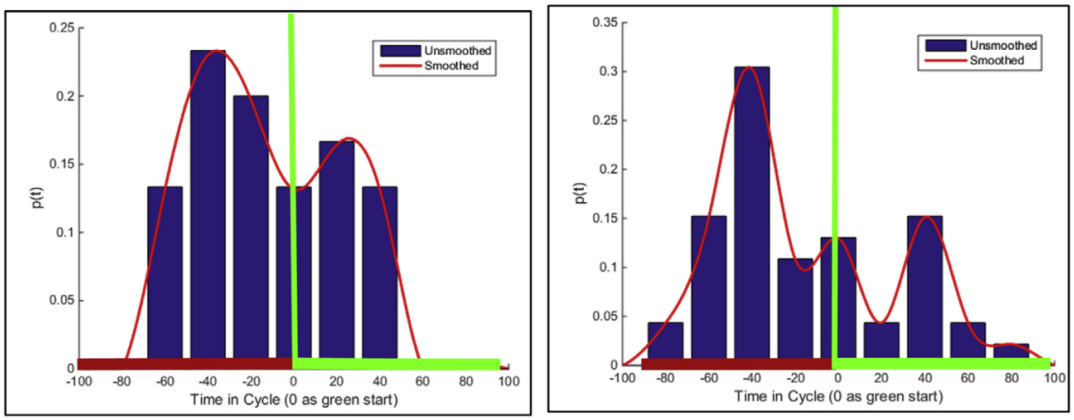


Figure 5.22 Illustration of Investigated Intersection.



(a) Sample CV Trajectories



(b) Time Dependent factor for 11AM-12 PM period (left) and 6 PM-7PM period (right)

Figure 5.23 Illustration of CV Trajectories (a) and time Dependent Factor (b) for EB through Movement.

For validation purpose, hourly volumes were also manually collected for two days, i.e., 04/25/16 and 04/26/16, from 11:00 AM to 7:00 PM. Using the measured volumes, we calculated the penetration rates of CVs, shown in Figure 5.24. Overall, the penetration rates ranged from 3% to 12%, varying over the selected periods. The rates also varied substantially at different approaches, with lower CV penetration rates at the EB and WB approach, i.e., the main approaches, and higher rates at the SB approach, a minor approach. This variation could be due to that the SB approach connects to residential areas close to the University of Michigan that would have larger population of participants of the SPMD project. The observed volumes were then used for comparing with the estimated volumes, with results shown in Figure 5.25. The three cases are shown in three sub-figures, respectively. In the figure, the yellow bars show the estimated volume, and the blue bars show the observed volumes, both in units of vehicle per hour per lane (vphpl). Substantial different traffic patterns exist in the three cases. For example, clear afternoon peak existed in both EB and SB cases, but not in WB case. Regarding the estimation, the estimated volumes are generally closed to the observed volumes for all the three cases. To further quantify the accuracy, we calculated the Mean Absolute Percentage Error (MAPE) for the estimation based on the following formula, indicated as well in the figure.

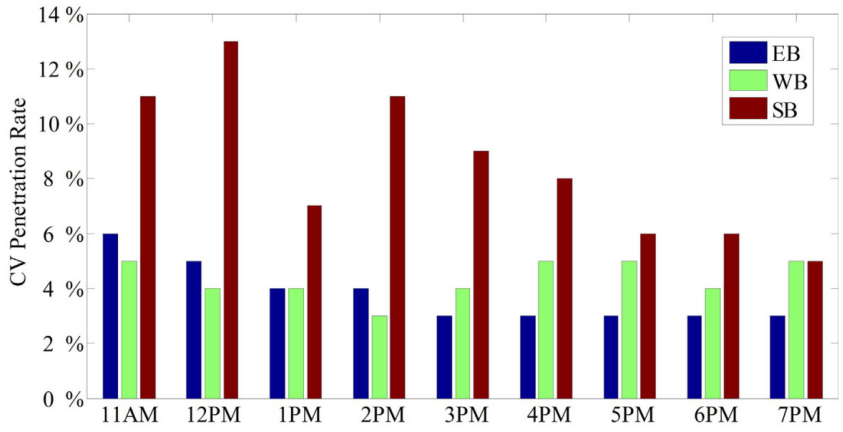
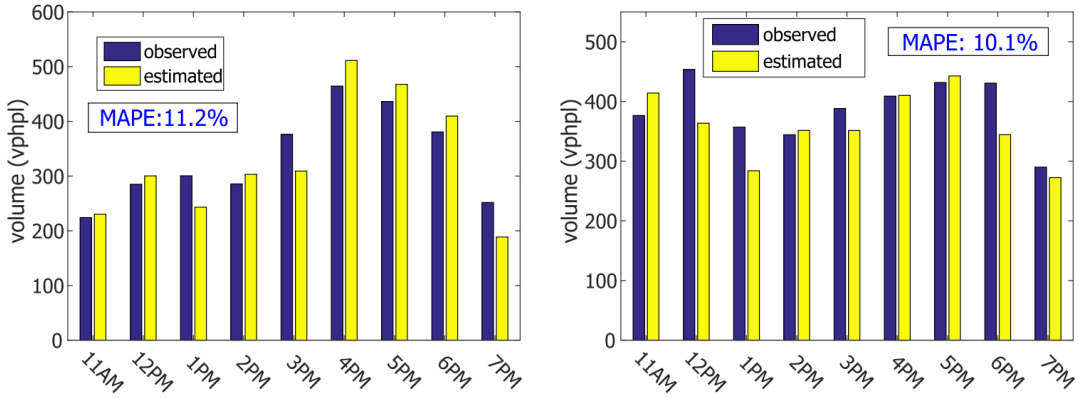
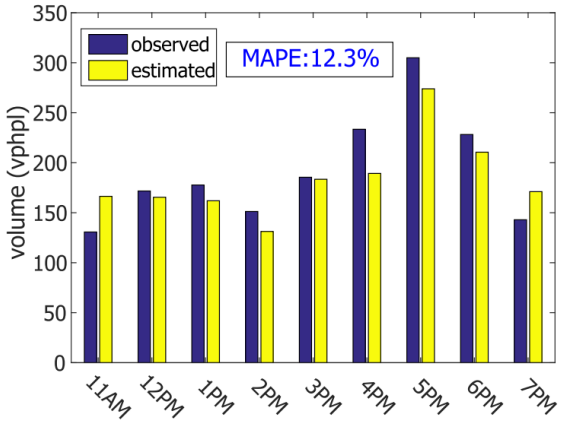


Figure 5.24 CV Penetration Rates Over Time of Day.



(a) EB-Through Movement (b) WB-Through Movement



(c) SB-Through and Left-Turn Movement

Figure 5.25 Comparison Between Observed Volume with Estimated Volume.

5.2.4 Real-time Adaptive Signal Control

In this section, a real-time adaptive traffic control algorithm is proposed with low penetration of connected vehicles. Existing studies showed that although minimum required penetration rates vary from different applications, but typically 20%-30% penetration rate is necessary (Day and Bullock, 2016). If the critical penetration rate cannot be reached, then data from traditional sources (e.g. loop-detectors) need to be added to improve the performance (Feng et al., 2016). Despite substantial efforts

in investing and developing CV technologies in the past decade, over the next ten years or longer, the CV penetration rate is expected to remain at a low level. Therefore, optimizing traffic signals with low penetration rates of CVs (< 10%) is important and will make an immediate impact on the state-of- the-practice.

Figure 5.26 shows the CV trajectories in one lane under 10% penetration rate with a demand level of 700 veh/h/ln. The trajectories represent the raw data used in this paper. It shows that some CVs passed the intersection without stop while others stopped in the queue for a red signal. Some of the vehicle trajectories are only partial because of lane changes. Note that during most of the cycles only one or two CVs were observed and during some cycles, there was no CV.

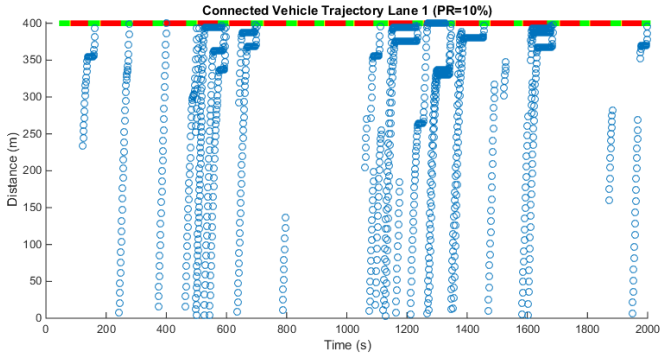
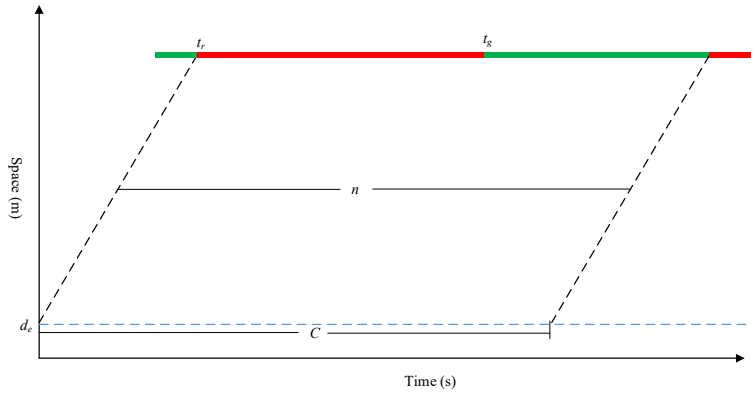
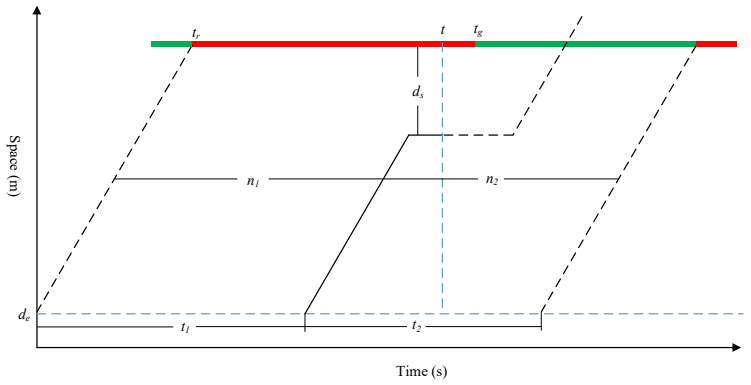


Figure 5.26 Illustration of CV trajectory under 10% penetration rate.

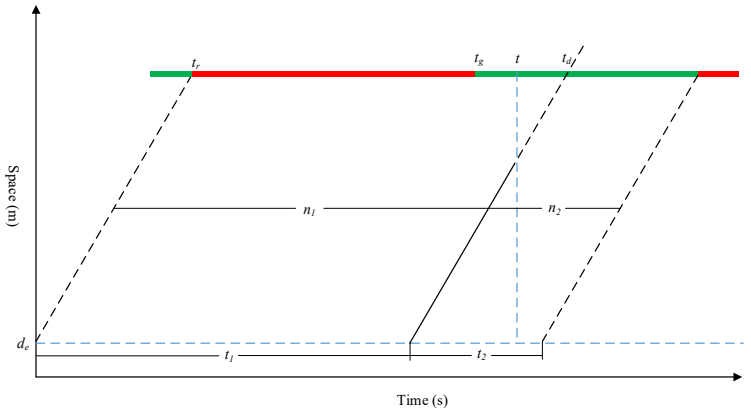
The basic idea of using limited CV trajectories to estimate total vehicle delay is to utilize critical CV information. Critical CVs are defined as the last stopped CV and the first non-stopped CV. The last stopped CV provides a lower boundary of queue length while the first non-stopped CV provides an upper boundary because the queue has to be fully discharged before the arrival of the non-stopped CV. For those cycles that don't have any CV observed, an average hourly volume is used to generate vehicle arrivals and departures for delay estimation. The hourly volume can be estimated from the aggregation of historical CV trajectory data (Zheng and Liu, 2017). We assume vehicle arrival follows Poisson process with mean arrival rate λ. The cumulative number of arrivals during time t is expressed as $N(t)$~Poisson(λt). Four cases are identified according to the existence of observed CVs as shown in 5.27.



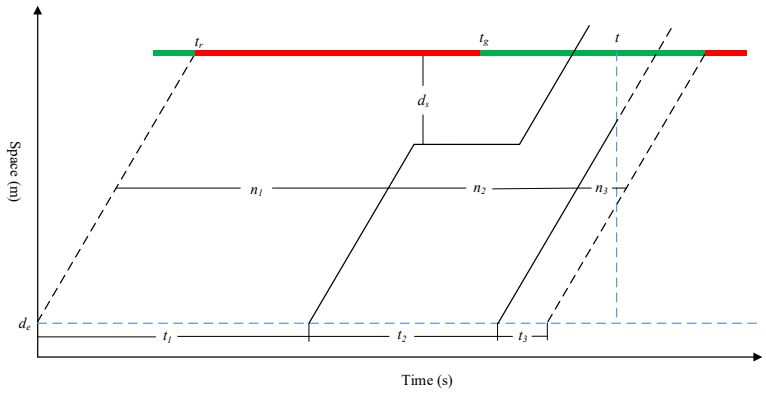
(a) No CV



(b) Only Stopped CV



(c) Only Non-stopped CV



(d) Both Stopped and Non-stopped CV

Figure 5.27 Four Scenarios Based on Critical CV Trajectory.

If multiple stopped and non-stopped CVs are observed within one cycle, only the last stopped CV and the first non-stopped CV are utilized because they represent the critical information. More details about the estimation models can be found in (Feng et al., 2018).

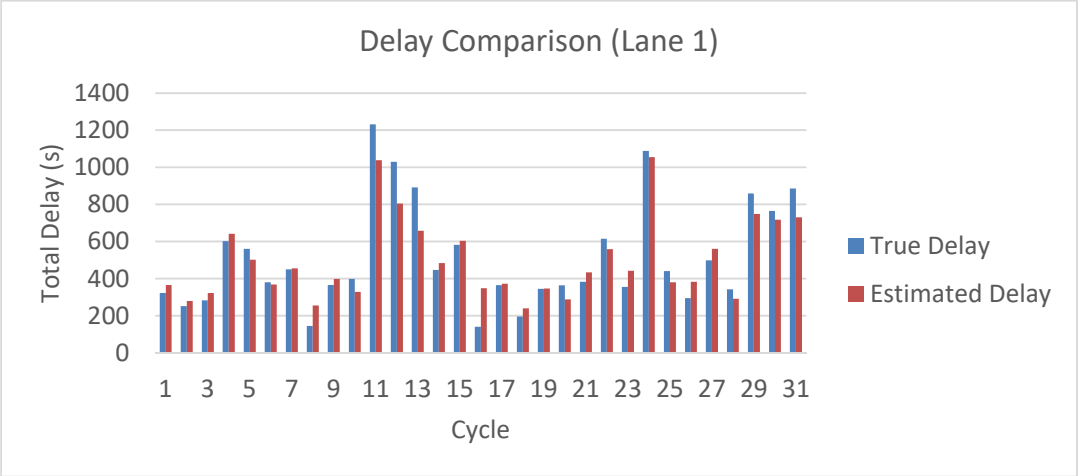
To evaluate the proposed delay estimation model, a VISSIM simulation model of Plymouth and Huron intersection was run for one hour and all vehicle trajectories were recorded and served as the ground truth. The traffic signals were under actuated control so that the cycle lengths and phase splits changed

over time. Figure 25.28 shows the comparison of estimated total vehicle delay and true vehicle delay of Phase 6 by lane with 10% penetration rate. There were total 31 full cycles operated within one hour.

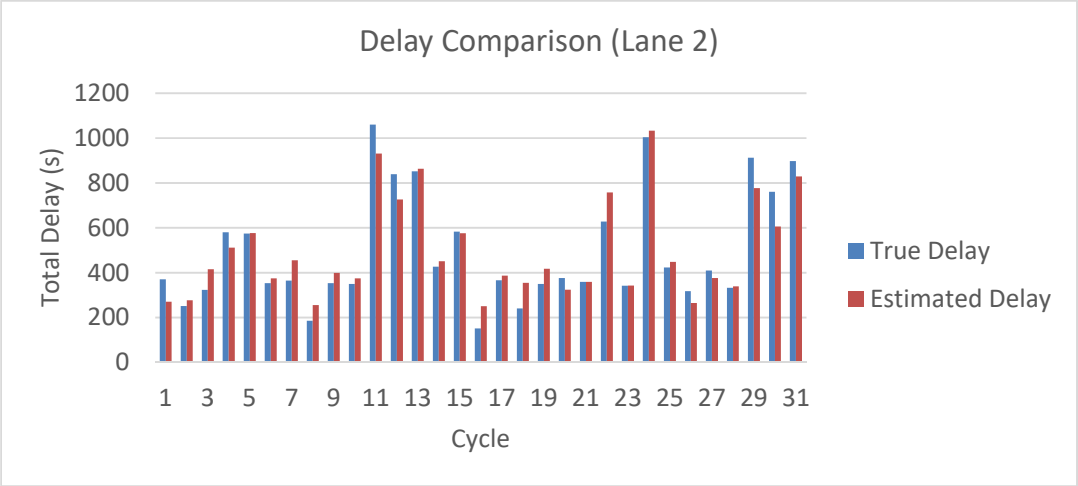
To further quantify the accuracy, we calculated the Mean Absolute Percentage Error (MAPE) for the estimated delay using the following equation

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|D_i^e - D_i^t|}{D_i^t} \quad (8)$$

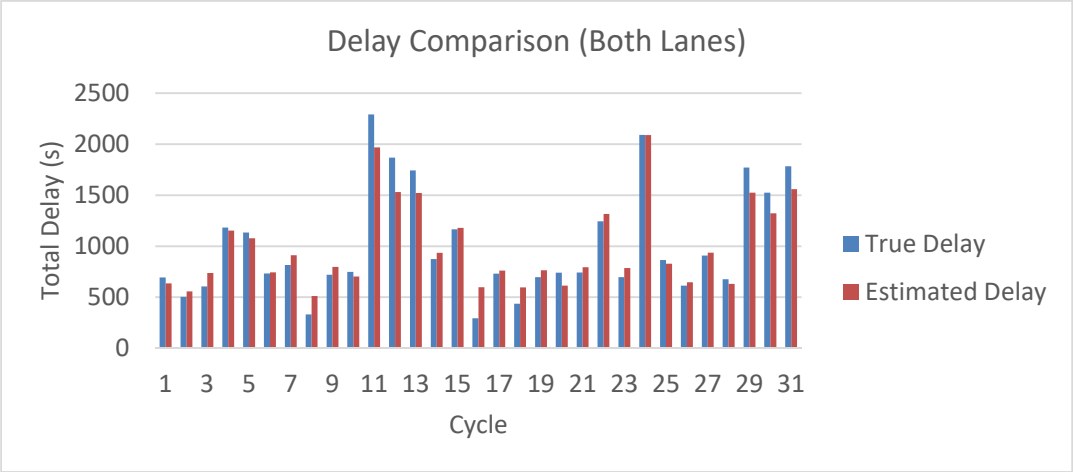
Where *N* is the total number of cycles, D_i^e is the estimated vehicle delay of cycle *i*, and, D_i^t is the true vehicle delay of cycle *i*. Under 10% penetration rate, the MAPE for Lane 1 and Lane 2 are 18.99% and 14.56% respectively. If two lanes are combined together, the MAPE for Phase 6 is 14.30%. We also tested the model under 0% penetration rate, under which only hourly volume was used to generate vehicle arrivals (always in Case 1 because of no CV). The MAPE for Lane 1 and Lane 2 are 32.60% and 28.65% respectively. If two lanes are combined together, the MAPE for Phase 6 is 30.49%. The result indicates that using hourly volume can't accommodate cycle-by-cycle traffic demand fluctuations well. From Figure 26.18 (c), it can be seen that the vehicle delay of each cycle varied from less than 500 veh·s to over 2000 veh·s. However, estimation from only 10% CV's data reduced the MAPE significantly, from more than 30% to less than 15% percent. It suggests that only a few critical CV trajectories are needed to estimate vehicle delay to a relatively accurate level.



(a) Lane 1



(b) Lane 2



(c) Combination of both lanes

Figure 27.28: Estimated Vehicle Delay Under 10% Penetration Rate.

The adaptive control algorithm is adapted from the DP based signal optimization algorithm introduced in Section 5.2.1. Total delay minimization is chosen as the objective function. To test the proposed models, a SIL simulation framework is designed and implemented with VISSIM microscopic simulation software. The SIL simulation architecture is shown in Figure 5.29.

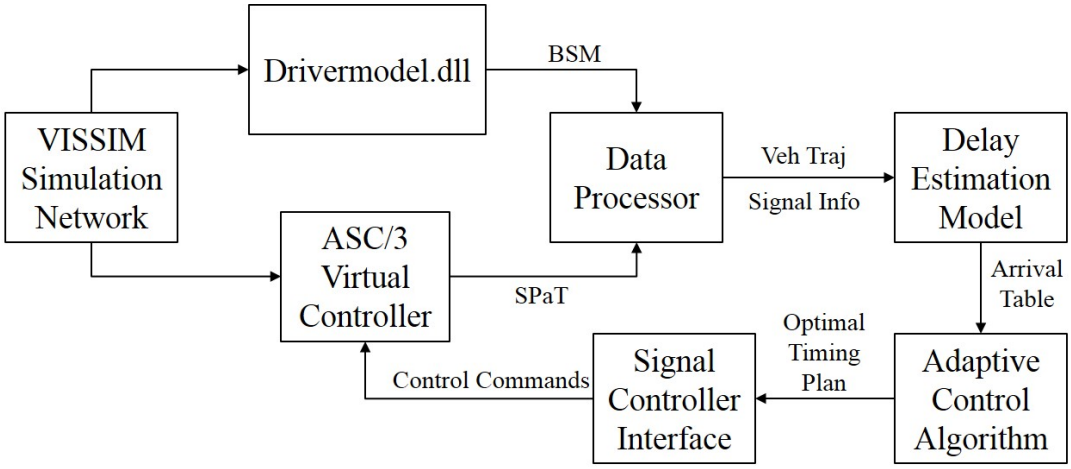


Figure 5.29: Software-in-the-loop Simulation Architecture.

CVs in VISSIM simulation network generate Basic Safety Messages (BSMs) at a frequency of 10Hz and broadcast to the Data Processor application. This application also requests Signal Phasing and Timing (SPaT) data from the Econolite ASC/3 virtual controller. Processed CV trajectory and signal information are then sent to the Delay Estimation Model. This module generates the arrival table and sends to the Adaptive Control Algorithm, which is responsible for producing optimal signal timing plan with the objective to minimize total vehicle delay. The optimal signal plan will be converted into a series of control commands by the Signal Controller Interface application and control virtual signal controllers in VISSIM. A real-world intersection at Huron Pkwy and Plymouth Rd in Ann Arbor, Michigan is modeled in VISSIM 9. The intersection geometry and signal phasing are shown in Figure 5.30.

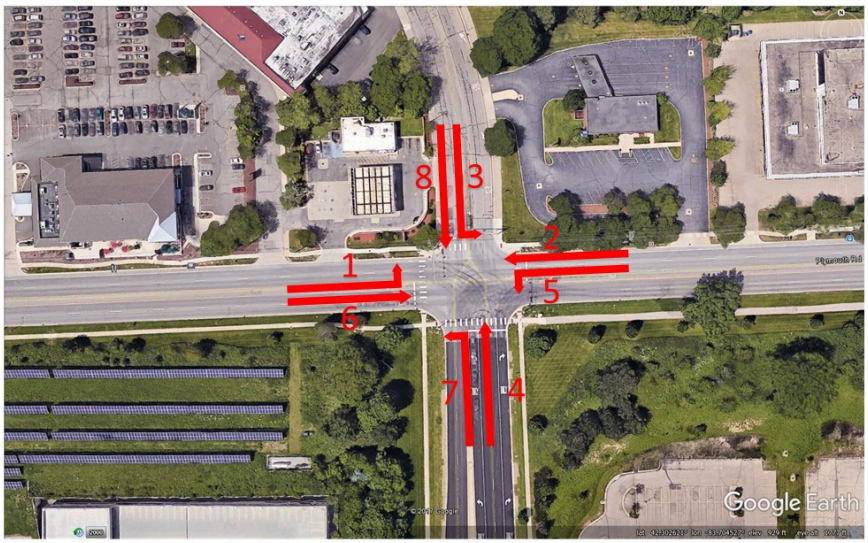


Figure 5.30: Geometry and Signal Phasing at Huron Pkwy & Plymouth Rd Intersection.

Since the delay estimation algorithm generates individual vehicle arrival times, an arrival table can be easily constructed and served as the input to the adaptive control algorithm. Two scenarios with two different demand levels and four penetration rates are evaluated. Scenario 1 assumes that the estimated hourly volume of each phase (or average arrival rate λ) is accurate. Scenario 2 assumes the estimated hourly volume of each phase has 10% error, which is more realistic based on field data. In scenario 2, we add 10% of demand on phase one to four and deduct 10% of demand on phase five to eight. The objective of such adjustment is to maximize the disturbance on the signal timing. Two demand levels are considered as medium (critical v/c ratio 0.82) and congested (critical v/c ratio 0.93)

traffic conditions. Four penetration rates under evaluation are: 10%, 5%, 2% and 0%. Under 0% penetration rate, the adaptive control basically becomes a fixed time signal plan, which is generated by the hourly volume (always Case 1 in delay estimation algorithm). The traffic demands used in each scenario are summarized in 5.7. Note that the estimated hourly volume with 10% error is only used in the delay estimation model. The vehicle inputs in the VISSIM are the same for the two scenarios, which is the actual hourly volume.

Table 5.7 Traffic Demands of Each Phase Under Two Scenarios and Two Demand Levels

Unit: veh/h/ln	P1	P2	P3	P4	P5	P6	P7	P8
Medium Demand (Scenario 1)	187	675	133	450	150	656	150	333
Medium Demand (Scenario 2)	206	742	146	495	135	591	135	300
Congested Demand (Scenario 1)	212	765	167	525	170	744	175	417
Congested Demand (Scenario 2)	233	841	182	577	153	670	157	375

A total duration of 3900s is executed in VISSIM simulation for each scenario, each demand level, and each penetration rate, with 300s of warm-up period and 3600s of data collection time. To capture the stochastic pattern, each simulation run is repeated with 5 different random seeds. The results are compared to a well-tuned fully actuated control, in which the minimum green time, maximum green time, yellow interval and all red clearance interval are set to be the same as in the adaptive control algorithm. The unit extension time of the actuated control is set to 1.6s, which is obtained by the recommendations from Signal Timing Manual. Table 5.8 and Table 5.9 shows the total vehicle delay under two demand levels.

TABLE 5.8 Total Vehicle Delay in Seconds under Medium Demand Level

Random Seed	1	2	3	4	5	Average (SD)	Delay Reduction
Scenario 1: Accurate hourly volume estimation							
10% PR	143336	152534	135818	151338	137554	144311 (7674)	5.23%
5% PR	148165	157135	141530	158741	149372	150988 (7034)	0.84%
2% PR	168963	190877	152779	178334	168224	171835 (14046)	-12.84%
Actuated	145736	162606	150933	158352	143770	152279	N/A

						(8070)	
Scenario 2: 10% hourly volume estimation error							
10% PR	144404	155736	143002	155517	149726	149677 (5983)	1.71%
5% PR	157791	168744	146392	159259	151568	156750 (8447)	-2.94%
2% PR	164093	182495	145614	170820	164004	165405 (13386)	-8.62%
Actuated	145736	162606	150933	158352	143770	152279 (8070)	N/A

Note: SD = Standard Deviation

TABLE 5.9 Total Vehicle Delay in Seconds under Congested Demand Level

Random Seed	1	2	3	4	5	Average (SD)	Delay Reduction
Scenario 1: Accurate hourly volume estimation							
10% PR	227684	248169	222959	260393	231441	238129 (15656)	16.33%
5% PR	240871	258387	222687	260856	231085	242777 (16692)	14.70%
2% PR	259532	281069	240524	280446	242579	260830 (19631)	8.35%
0% PR	327241	367273	288306	344282	261268	317674 (42731)	-11.62%
Actuated	256728	305282	279268	330017	251736	284606 (33074)	N/A
Scenario 2: 10% hourly volume estimation error							
10% PR	252124	282365	243068	279463	258485	263101 (17189)	7.56%

5% PR	267432	283013.	242671	271912	249347	262875 (16577)	7.64%
2% PR	270629	339032	254176	317639	281232	292541 (34897)	-2.79%
0% PR	346828	380832	356243	442983	313010	367979 (48470)	-29.29%
Actuated	256728	305282	279268	330017	251736	284606 (33074)	N/A

Note: SD = Standard Deviation

The following observations are made by analyzing the results:

1. When the penetration rate is 10%, the proposed model outperforms well-tuned actuated control in all cases. The total vehicle delay is decreased by 16.33% under congested demand level with accurate volume estimation. Under medium demand level with 10% volume estimation error, the vehicle delay is still reduced by 1.71%. As the penetration rate decreases, the total delay tends to increase.
2. The hourly volume estimated from historical data has a significant impact on the performance. Under same demand level and same penetration rate, the results with 10% volume estimation error are all worse than no error in volume estimation. This is because the algorithms are executed under very low penetration rates. It is common that no connected vehicle is observed within the entire cycle. Then the hourly volume serves as the only data for determining the phase duration.
3. Besides penetration rate, the absolute number of observed CV is also crucial to the performance of the algorithm. This explains why the algorithm performs better under congested demand level than medium demand level with same penetration rate. Under congested demand level with accurate volume estimation, even 2% penetration has a delay reduction of 8.35%. However, under medium demand level with accurate volume estimation, model performance with 5% penetration rate is almost the same as actuated control.
4. Vehicle delays with 10% and 5% penetration rates under congested demand level are similar, in both scenarios. This indicates that a few critical vehicle trajectories are enough to make an accurate estimation of vehicle delay. Higher penetration rate only receives marginal benefits.
5. When the algorithm is executed under 0% penetration rate, the adaptive control becomes a fixed time control. Because no CV trajectory is available, the control decision is made only based estimated hourly volume, which is a set value. The results under such conditions are significantly worse than other cases, which supports a well-accepted argument that fixed time control can't accommodate short time demand fluctuation, even if the average volume is accurate. Moreover, under congested demand level, the intersection under fixed time control may enter oversaturated condition easily due to demand fluctuation, and the delay increases significantly. On the other hand, actuated and adaptive control can handle the demand fluctuation better and prevent the intersection enter the oversaturated condition.

5.2.5 Semi-adaptive Traffic Signal Control

Crowdsourced vehicle trajectory data, e.g., from connected vehicles or ride-hailing service providers such as Uber in the U.S. or Didi in China, are increasingly available. These trajectory data would potentially revolutionize traffic signal operation. Different from conventional detector data, trajectory data could serve as a low-cost, continuous and reliable data source, which could advance conventional detector-based signal control to a detector-free signal control scheme. Utilizing trajectory data as the sole data source, this research proposes a closed-loop intelligent system for monitoring and control of traffic signals. The concept of the system process is shown in Figure 5.31. The iteration loop consists of totally four parts: sensing, evaluation, optimization, and implementation.

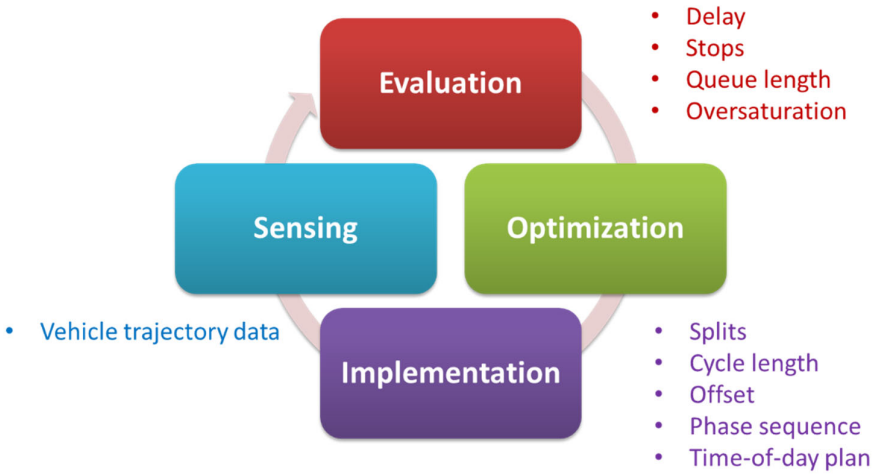


Figure 5.31 Closed-loop iterative evaluation-optimization

As a closed-loop process, the optimization and control implementation can be repeated periodically, e.g., once every week, depending on data availability. This introduces a new signal retiming process. Existing signal control systems can be categorized into two groups regarding retiming frequency. The first group is represented by fixed-time traffic signals. Since one could only rely on manual data collection, the sensing part is the bottleneck that restricts the iteration frequency of the loop. The signal timing parameters could not be updated in a timely fashion. The second group includes vehicle actuated control and adaptive signal control, in which signal phases can be adjusted dynamically in response to traffic variations. However, traffic engineers need to properly maintain detectors and correctly set a group of parameters to ensure good performance. Different from the two types of signal systems, here, the proposed system combines fixed timing signal with weekly parameter update. With the belief that well parameterized fixed timing control can accommodate main traffic patterns robustly, the system may not need to respond to traffic fluctuations frequently. Instead, it focuses on the demand change over longer terms due to seasonal change, road construction and so on, and updates signal control parameters weekly or biweekly.

Overall, the proposed system is an integrated platform to evaluate and optimize traffic signals. With vehicle trajectories as the sole input, the system can generate performance evaluation report, and calculate optimal parameters of signal timings. The performance evaluation will focus on detecting oversaturation and assessing signal coordination quality to capture key information of traffic condition in a road network. Visualization of the performance, e.g., in the form of time-space diagram, will also be provided so that results can be easily interpreted by traffic engineers. For signal optimization, a hierarchical optimization process is developed to optimize signal timing plan schedule, as well as cycle length, green split, and offset. The optimization process and control implementation can be repeated periodically. The propose system can be used either as a supporting tool for management agencies in

their daily signal operation, or to control traffic signal system directly with periodic updates of parameters.

Overall Framework

The overall optimization framework is shown in Figure 5.32. The framework consists of three main components: demand estimation, performance visualization and system optimization.

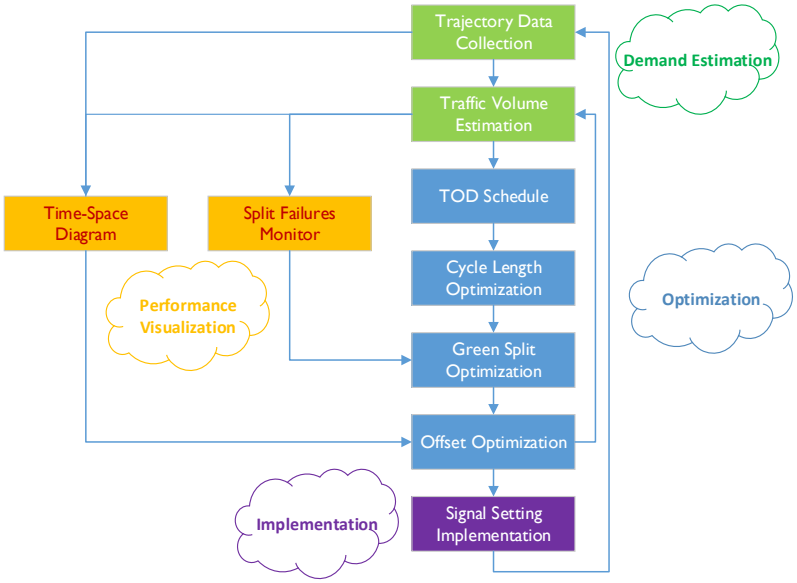


Figure 5.32 Overall design of the optimization system

Demand Estimation

The demand estimation was built upon the research of volume estimation using trajectory data presented in Section 5.2.3. The demand estimation component will utilize traffic signal status and vehicle trajectory data to estimate traffic volume, which is the key input for signal optimization. It was demonstrated that good accuracy can be achieved using vehicle GPS data with penetration rates of 1% - 6 %. Based on the estimated traffic demand, performance evaluation can be conducted with many well-established traffic engineering tools.

Performance Visualization

The performance evaluation will focus on detecting oversaturation and assessing signal coordination quality to capture key information of traffic condition in a road network. The estimated demand information will be used to visualize the performance of traffic signal system, so that results can be easily interpreted by traffic engineers. This component consists of two convenient tools: split failure monitor and time-space diagram. The split failure monitor shows intersection status regarding whether the allocated green times suffice to serve traffic demands for each phase, and identify oversaturation at intersections. The time-space diagram shows how vehicles travel through intersections along a corridor to evaluate signal coordination quality.

Parameter Optimization

The parameter optimization component consists of three modules to optimize different signal settings: 1). time of day (TOD) schedule optimization, 2). cycle length and offset optimization, and 3). green split optimization. The overall optimization will be performed in the following fashion: first, TOD schedule will be optimized by applying clustering technique. Then, within each TOD interval, cycle length and offsets will be optimized considering coordination between adjacent intersections within the network.

Lastly, green splits will be optimized for each intersection to reduce oversaturation as well as to balance green/capacity ratio across conflict movements.

Detailed optimization methodologies can be found in (Zheng et al., 2019).

Case Study

The proposed system was implemented to optimize signal timings in the City of Jinan, China. Data from Didi drivers are used as the only input. Note that only the trajectories of Didi vehicles with at least one passenger on board were selected. In this way, we could exclude trajectories with unusual vehicle stops that are not caused by driver detouring for searching customer.

This case study was to optimize signal timings at JinShi Road, an important corridor in the City of Jinan. Six intersections along the road were selected for investigation. The illustration of the selected segment is shown in Figure 5.33.

Figure 5.33 Illustration of intersections on JingShi

Based on detector data that are available from one intersection, we calculated the penetration rate of Didi vehicles for every 30-min interval. As shown in Figure 5.34, the penetration rate varied from 2% to 6%, with the peak of penetration rate occurred around 9:00 am.

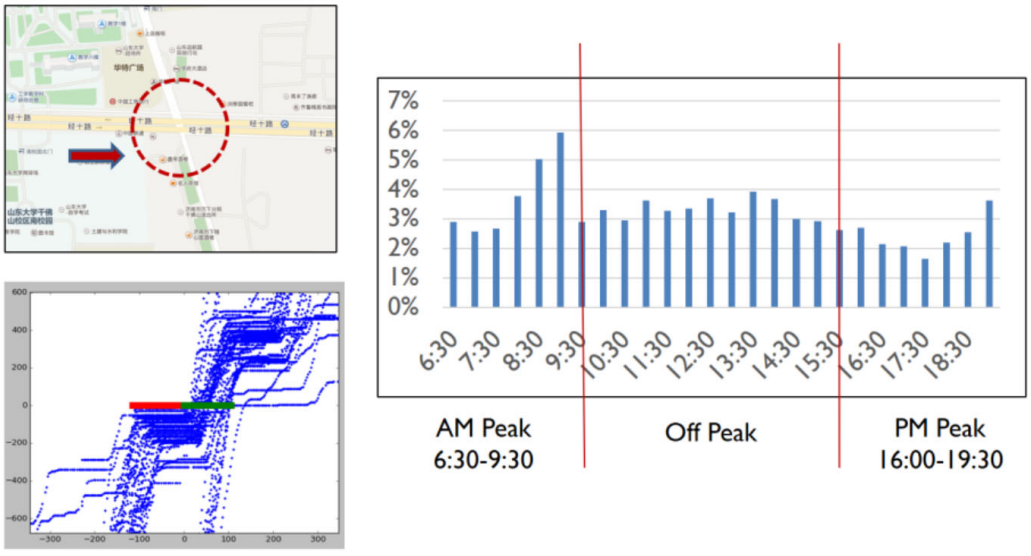


Figure 5.34 Example of penetration rate of occupied Didi vehicles

Traffic volumes were estimated based on the trajectory data. The validation of the estimation is performed by comparing the estimated volume with detector data. The result is shown in Figure 5.35. Overall, the mean absolute percentage error is 8.6%.

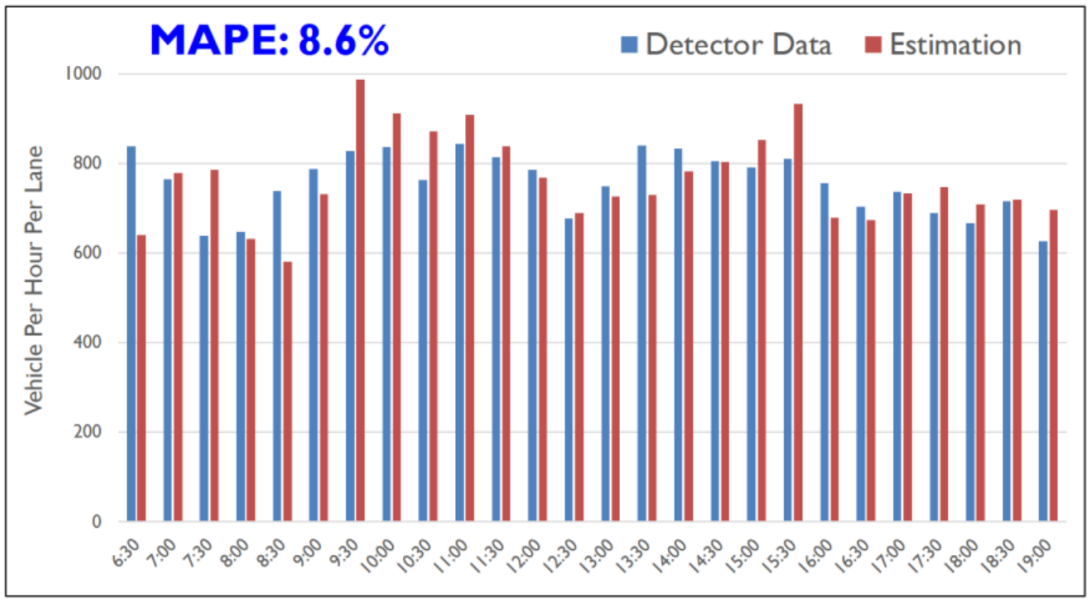


Figure 5.35 Comparison between estimated volumes with observed volumes from detector data

Using the estimated volume, TS-diagrams were constructed to evaluate coordination performance across six intersections. An example is illustrated in Figure 5.36. The horizontal axis represents time and the vertical axis represents distance. The trajectories are aggregated over the study time period, for weekday off-peak hours. The trajectories shown here are vehicles travelling downward, and those flat trajectories indicate vehicle stopping in queues. The left figure shows the raw trajectory of Didi vehicles, while the right figure shows the estimated trajectories of all traffic, which include both Didi vehicles and regular vehicles.

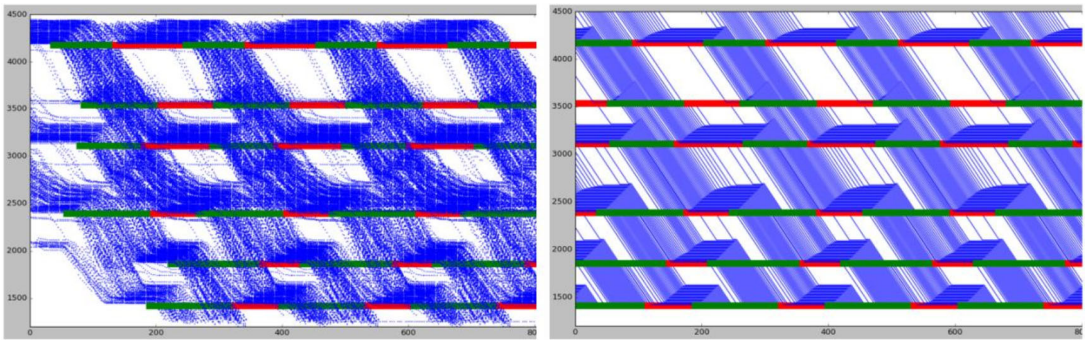


Figure 5.36 Time space diagram with raw trajectory (left) and estimated trajectory (right) Field Implementation and Optimization Results

Parts of the optimization algorithms are implemented at the selected corridor. There are two restricting characteristics of the selected segment: 1. Due to a large pedestrian traffic crossing the main street and long crossing time, the cycle length needs to be fixed as a relatively large value (220s for peak hours, and 210s for non-peak hours), to ensure minimum green durations for side-street phases with small g/c ratio. 2. Due to reversible lanes used for left turn traffic at the main-street, the left-turn phases have to be served as lead-phases at the main-street, i.e., phase sequences are fixed. Therefore, cycle length and phase sequence were not changed. For the selected segment, we optimized three TOD plans during the daytime, including AM peak, PM peak, and off-peak timing plan. To evaluate the results of the changes, we calculated the relative change of three performance indexes: arterial delay, total delay and the number of spillover vehicles during red signal, before and after the changes. In details:

1. Total delay is the average delay time of all Didi vehicles at all movements of an intersection, and is further averaged across all intersections.

2. Arterial delay is the vehicle delay for traveling through the corridor, averaged across the two opposite directions.

3. The number of spillover vehicles during red light is the number of Didi vehicles stopped at the intersection area during red light. These spillover vehicles during red light might block crossing traffic from the side street and even cause intersection gridlock. Therefore, the spillover vehicles are a particular concern during peak hours. The results regarding the relative changes of the performance indexes are summarized in Table 5.10.

Table 5.10 Relative change of performance indexes for JingShi Road

	AM Peak	**PM Peak**	**Off-Peak**
Arterial Delay	+3.53%	-21.87%	-5.85%
Total Delay	-10.73%	-10.94%	-6.32%
# of spillover vehicles during red	-80.1%	-88.2%	/

As shown in the table, the majority of the performance improved, with reduction of delay and number of spillover vehicles during red. The largest reduction exists for arterial delay during PM peak, by 21.87%, with the rest of delay reduction ranging from 5% to 11%. The number of spillover vehicles during red reduced significantly with 80.1% reduction in AM Peak period, and 88.2% reduction in PM Peak period. However, for AM peak, arterial delay increased slightly, with 3.53%. This is likely because, in the AM peak plan, the optimized timing plan was slightly adjusted to further reduce spillover vehicles during red. The reduction of spillover vehicles during red would not be beneficial to the main-street traffic, but helpful to facilitate crossing traffic.

The reduction of spillover vehicles during red signals is illustrated in Figure 5.37. With the original signal timing plan, spillover vehicles indicated by those flat trajectories near the intersection area can be clearly seen in the left figure. With the optimized signal timing plan, few spillover vehicles were observed in the right figure. Overall, the improvements are encouraging.

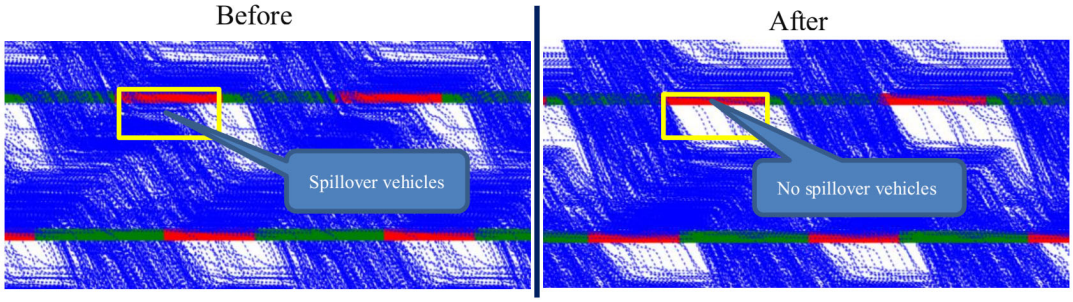


Figure 5.37 Illustration of reducing spillover vehicles during red signal

5.3 Deployment and Field Experiment

In this task, we implemented the methodology developed in Section 5.2.5 to five intersections at Plymouth Rd corridor. This experiment includes data collection, data processing, generation of performance measures, and generation of optimal signal timing.

Ten days of trajectory and signal data from weekday PM peak hour (3:00pm – 6:00pm) are queried from the Safety Pilot database and used for data analysis. Two types of data are used: BSM and SPaT. The BSM data contain timestamp, vehicle ID, vehicle location (longitude, and latitude) and vehicle speed. The SPaT data contain timestamp, intersection ID, phase status and duration. Both types are collected at the frequency of 10Hz. A corridor of five intersections on Plymouth Rd is included in the data analysis. A corridor lane phase mapping is constructed to map the trajectories data and signal timing to the correct vehicle movements as shown in Figure 5.38.

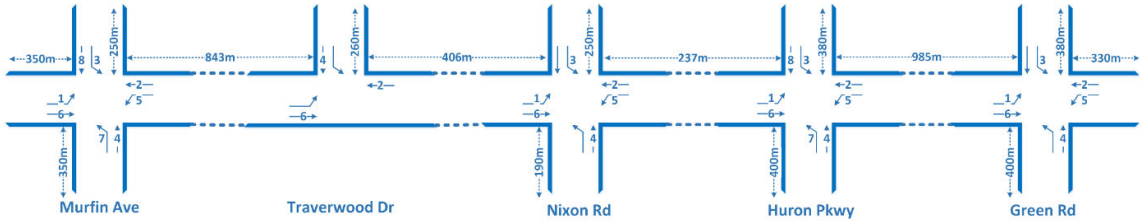


Figure 5.38 Lane Phase Mapping of the Plymouth Corridor

Figure 5.39 and Figure 5.40 show aggregated vehicle trajectories passing the intersection over the ten days at Traverwood and Nixon intersections respectively. Blue traces indicate through movement trajectories, red traces indicate left turn movement trajectories, and green traces indicate right turn movement trajectories. Table 5.11 and Table 5.12 show the total number of trajectory traces for each movement. It can be seen that through movement on the major road (westbound and westbound) have the majority number of traces while other movements have less traces. This is consistent with the traffic volume at each movement. In general, about 50 trajectory traces are necessary for estimating the volume and optimizing traffic signal parameters accurately. For those movements that have less traces, the estimation accuracy may decrease. However, those movements also have less impact on the total traffic flow.

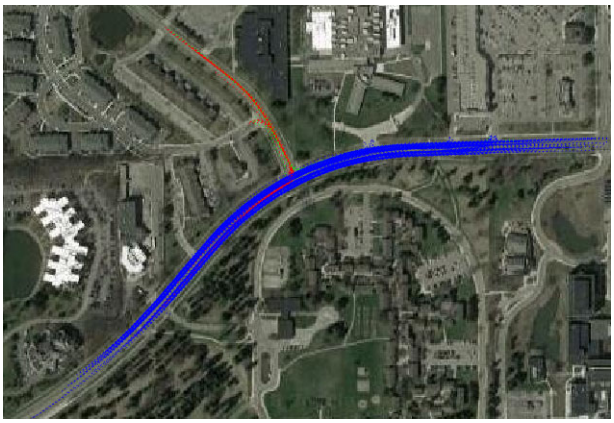


Figure 5.39 Aggregated Trajectory Traces at Traverwood and Plymouth Intersection

Table 5.11 Number of Trajectory Traces for Each Movement at Traverwood and Plymouth Intersection

	Through	Left	Right

EB	160	21	-
WB	181	-	10
NB	-	-	-
SB	-	6	1

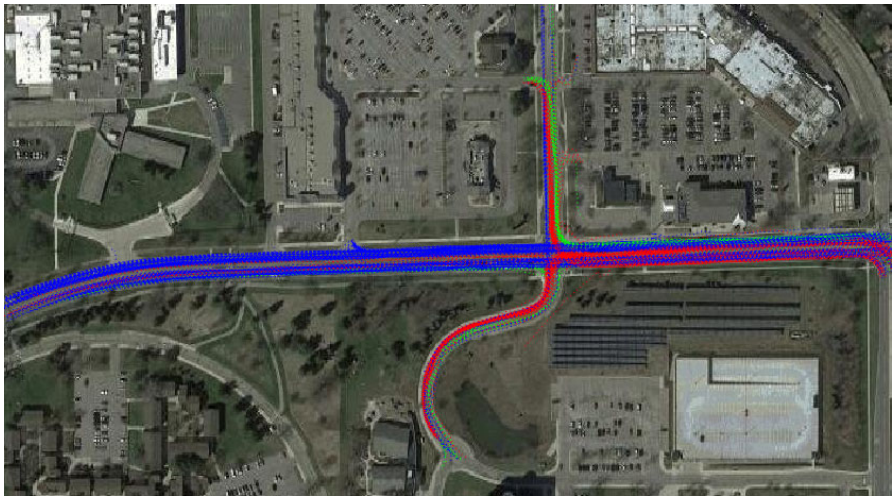


Figure 5.40 Aggregated Trajectory Traces at Nixon and Plymouth Intersection

Table 5.12 Number of Trajectory Traces for Each Movement at Nixon and Plymouth Intersection

	Through	Left	Right
WB	107	9	14
EB	153	63	21
NB	8	5	14
SB	14	39	0

Figure 5.41 and Figure 5.42 show a period (10 mins) of SPaT data for the Traverwood and Nixon intersections. Different colors represent different phases. The on/off time and durations of each signal phase can be clearly seen from the figures. Note that we only include green and red signal status in the analysis. The yellow interval is also collected but not used directly. The yellow interval is represented by effective green time and effective red time. This is a common practice in signal optimization so that signal status can be treated as a binary variable. Usually the yellow interval is divided into two portions. The first portion is considered as green and the second portion is considered as red. It is also consistent with vehicle behaviors in the real-world.

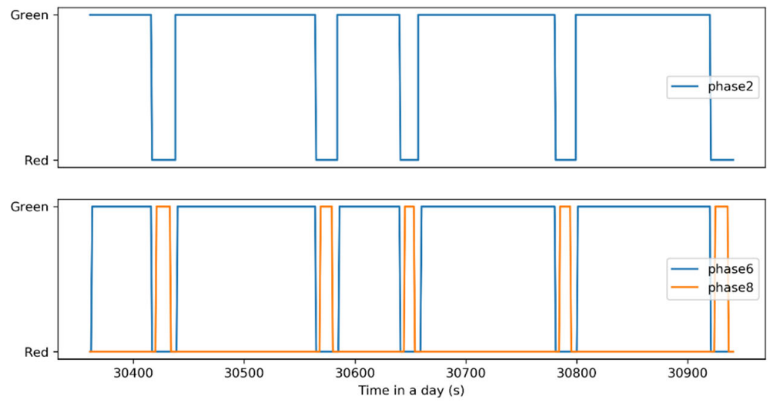


Figure 5.41 Phase Switch and Duration at Traverwood Intersection

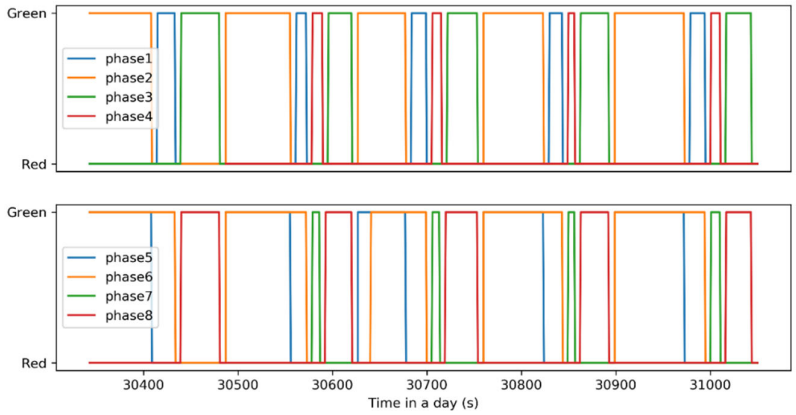
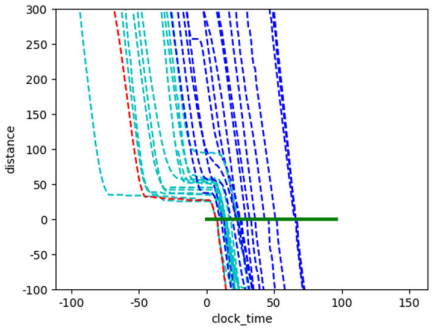
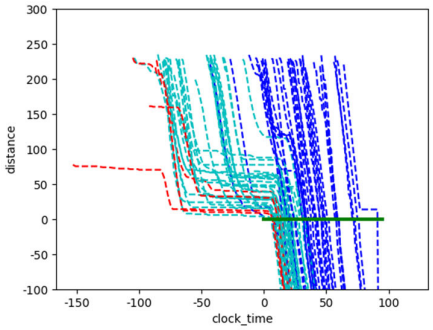


Figure 5.42 Phase Switch and Duration at Nixon Intersection

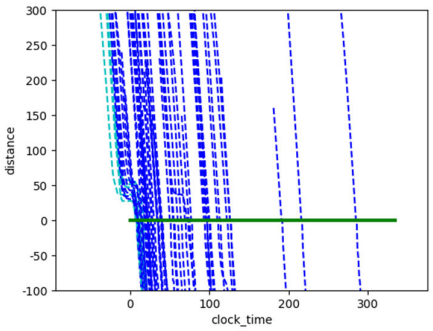
Combining BSM and SPaT data, the time space diagrams can be drawn to visualize the vehicle behaviors at the intersections as shown in Figure 5.43 and Figure 5.44. The two figures show the eastbound and westbound through movements, which represent the majority of traffic. The x-axis is the time in a signal cycle and zero is the start of green signal. First, the trajectory data are grouped by signal cycle and the absolute timestamp is then converted into its relative time offset in the cycle. The blue traces mean that the vehicle passed the intersection without stop, the green traces mean that the vehicle passed the intersection with one stop, and the red traces mean that the vehicle passed the intersection with two stops. The green bar at zero distance on y-axis represents the green duration. Since it is an aggregated diagram, the length of the green bar indicates the maximum green time observed duration the data collection period. It can be seen from the figures that at Nixon intersection, more than half of the vehicles need to stop once and some of them need to stop twice. At Traverwood intersection, most of the vehicles can pass the intersection without stop. One reason is that Traverwood is a T-intersection so that most of the green time is allocated to the eastbound and westbound through movement (major street). Nixon is an eight phase intersection with higher volume on side street. However, the multiple stops at Nixon implies potential room for improvement through better signal progression.

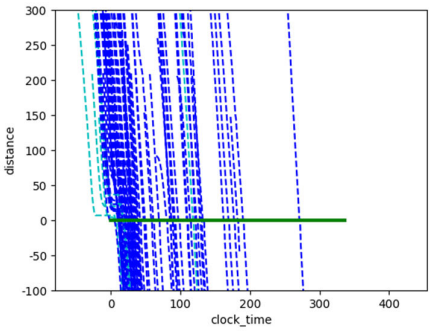




Eastbound Westbound

Figure 5.43 Time Space Diagram at Nixon Intersection





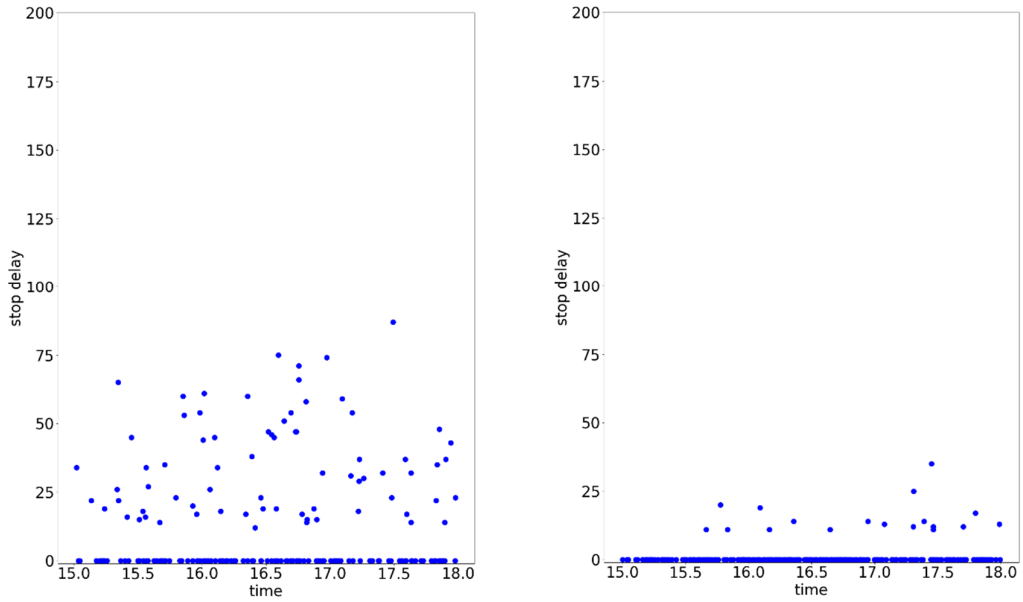
Eastbound Westbound

Figure 5.44 Time Space Diagram at Traverwood Intersection

Figure 5.45 shows the delay measurements at the two intersections. Delay measurements are derived directly from the time space diagram. It is calculated as the actual travel time minus the free flow travel time. Similar to the time space diagram, vehicle delay at Nixon intersection is much higher than the delay at Traverwood intersection.

Based on the analysis on the time space diagrams and delay diagrams of all the intersection, a new signal timing plan is optimized for the corridor with the objective to reduce total delay. The new plan keeps the signal phasing, sequence and transition time unchanged, but adjusts the green splits and offsets at each intersection. Figure 5.46 shows the optimized signal coordination plan for the five intersections. Due to the vehicle travel speed and distances between intersections, it is difficult to provide two-way progression. The new signal timing provides better progression at westbound (Green -> Murfin) direction. For westbound (Murfin -> Green), vehicles have to stop at the Nixon intersection once. It is mentioned above that the optimization is based on delay minimization. Since the westbound has higher traffic volumes for the through movements during afternoon peak hours, providing better progression on westbound has more benefit in reducing total delay.

Table 5.13 shows the complete signal timing plan for all the intersections. A uniform cycle length of 12s is applied to all the intersections. The offsets of each intersection are: 0s, 36s, 47s, 31s, and 91s.



Nixon Eastbound

Traverwood Eastbound

Figure 5.45 Delay Diagram at Nixon and Traverwood Intersections

Table 5.13 Complete Signal Timing Plan

Intersection	ID	Start	Duration	ID	Start	Duration	ID	Start	Duration	ID	Start	Duration
Murfin	2	0	60	6	0	60	1	60	12	5	0	12
Traverwood	2	0	90	6	0	90						
Nixon	2	0	54	6	0	48	1	54	12	5	48	18
Huron	2	12	48	6	0	48	1	0	12	5	48	12
Green	2	12	42	6	0	42	1	0	12	5	54	12
Intersection	ID	Start	Duration	ID	Start	Duration	ID	Start	Duration	ID	Start	Duration
Murfin	3	72	12	7	72	12	4	84	18	8	84	18
Traverwood	3	90	30									
Nixon	3	66	30	8	66	30	4	96	15	7	96	15
Huron	3	60	30	7	60	30	4	90	15	8	90	15
Green	3	66	18	8	66	18	4	84	30	7	83	30

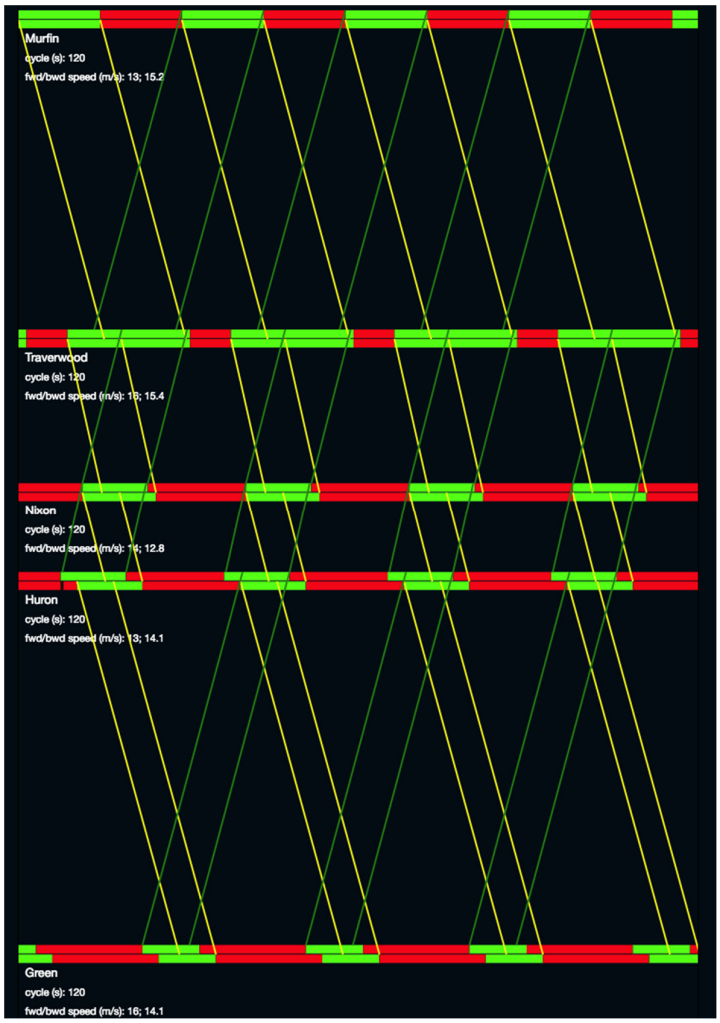


Figure 5.46 Optimized Signal Coordination Plan

The overall implementation process described in this section is summarized in Figure 5.47.

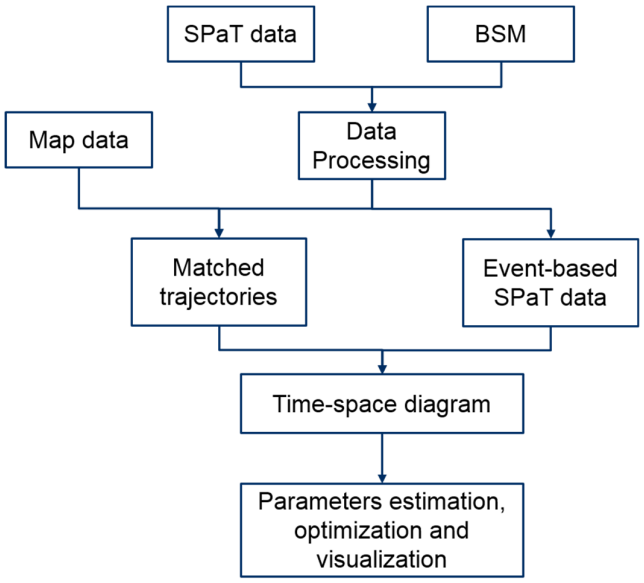


Figure 5.47 Overall Implementation Process

5.4 Publications Under Task 5

[1] Feng, Y., Yu, C. and Liu, H.X., 2018. Spatiotemporal Intersection Control in a Connected and Automated Vehicle Environment. *Transportation Research Part C: Emerging Technologies*, *89*, pp.364-383.

[2] Yu, C., Feng, Y., Liu, H.X., Ma, W. and Yang X., 2018. Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections. *Transportation Research Part B: Methodological,* 112, pp. 89-112

[3] Zheng, J. and Liu, H.X., 2017. Estimating traffic volumes for signalized intersections using connected vehicle data. *Transportation Research Part C: Emerging Technologies*, *79*, pp.347-362.

[4] Feng, Y., Zheng, J. and Liu, H.X., 2018. A Real-time Detector-Free Adaptive Signal Control with Low Penetration of Connected Vehicles. *Transportation Research Record*. Vol. 2672(18) 35–44

[5] Zheng, J., Sun, W., Huang, S., Shen, S., Yu, C., Zhu, J., Liu, B., and Liu, H.X., 2019. Traffic Signal Optimization Using Crowdsourced Vehicle Trajectory Data. *Transportation Research Part C: Emerging Technologies (Under Review)*

References

PTV, 2015. PTV Vissim 9 user manual. *Karlsruhe, Germany*.

Barth, M. and Farrell, J.A., 1999. The global positioning system & inertial navigation. *McGraw-Hill*, *8*, pp.21-56.

Barth, M., Mandava, S., Boriboonsomsin, K., Xia, H., 2011. Dynamic ECO-driving for arterial corridors, in: 2011 IEEE Forum on Integrated and Sustainable Transportation System (FISTS), pp. 182–188. doi:10.1109/FISTS.2011.5973594

Rakha, H., Kamalanathsharma, R.K., 2011. Eco-driving at signalized intersections using V2I communication, in: 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 341–346. doi:10.1109/ITSC.2011.6083084

He, X., Liu, H.X., Liu, X., 2015. Optimal vehicle speed trajectory on a signalized arterial with consideration of queue. Transp. Res. Part C Emerg. Technol. 61, 106–120. doi:10.1016/j.trc.2015.11.001

Wu, X., He, X., Yu, G., Harmandayan, A., Wang, Y., 2015. Energy-Optimal Speed Control for Electric Vehicles on Signalized Arterials. IEEE Trans. Intell. Transp. Syst. 16, 2786–2796. doi:10.1109/TITS.2015.2422778

Feng, Y., Head, K.L., Khoshmagham, S., Zamanipour, M., 2015. A real-time adaptive signal control in a connected vehicle environment. Transp. Res. Part C Emerg. Technol. 55, 460–473. doi:10.1016/j.trc.2015.01.007

Goodall, N., Smith, B., Park, B., 2013. Traffic Signal Control with Connected Vehicles. Transp. Res. Rec. J. Transp. Res. Board 2381, 65–72. doi:10.3141/2381-08

He, Q., Head, K.L., Ding, J., 2014. Multi-modal traffic signal control with priority, signal actuation and coordination. Transp. Res. Part C Emerg. Technol. 46, 65–82. doi:10.1016/j.trc.2014.05.001

Lee, J., Park, B., Yun, I., 2013. Cumulative Travel-Time Responsive Real-Time Intersection Control Algorithm in the Connected Vehicle Environment. J. Transp. Eng. 139, 1020–1029. doi:10.1061/(ASCE)TE.1943-5436.0000587

Malakorn, K.J., Park, B., 2010. Assessment of mobility, energy, and environment impacts of IntelliDrive-based Cooperative Adaptive Cruise Control and Intelligent Traffic Signal control, in: 2010 IEEE International Symposium on Sustainable Systems and Technology (ISSST), pp. 1–6. doi:10.1109/ISSST.2010.5507709

Li, Z., Elefteriadou, L., Ranka, S., 2014. Signal control optimization for automated vehicles at isolated signalized intersections. Transp. Res. Part C Emerg. Technol. 49, 1–18. doi:10.1016/j.trc.2014.10.001

Lee, J., Park, B., 2012. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. IEEE Trans. Intell. Transp. Syst. 13, 81–90. http://dx.doi.org/10.1109/TITS.2011.2178836

Zohdy, I.H., Rakha, H.A., 2014. Intersection management via vehicle connectivity: the intersection cooperative adaptive cruise control system concept. J. Intell. Transp. Syst. 1–16. http://dx.doi.org/10.1080/15472450.2014.889918.

Sethi, S.P., Thompson, G.L., 2000. Optimal Control Theory. Springer-Verlag, New York.

Feng, Y., Yu, C. and Liu, H.X., 2018. Spatiotemporal Intersection Control in a Connected and Automated Vehicle Environment. *Transportation Research Part C: Emerging Technologies*, *89*, pp.364-383.

EPA, 2002. Methodology for Developing Modal Emission Rates for EPA's Multi-Scale Motor Vehicle and Equipment Emission System (No. EPA420- R-2–27).

Yeo, H., Skabardonis, A., Halkias, J., Colyar, J., Alexiadis, V., 2008. Oversaturated Freeway Flow Algorithm for Use in Next Generation Simulation. Transp. Res. Rec. J. Transp. Res. Board 2088, 68–79. doi:10.3141/2088-08

Newell, G.F., 2002. A simplified car-following theory: a lower order model. Transp. Res. Part B Methodol. 36, 195–205. doi:10.1016/S0191-2615(00)00044-8

Gipps, P.G., 1981. A behavioural car-following model for computer simulation. Transp. Res. Part B Methodol. 15, 105–111. doi:10.1016/0191-2615(81)90037-0

Sen, S., Head, K.L., 1997. Controlled Optimization of Phases at An Intersection. Transp. Sci. 31, 5–17.

Emmelmann, M., Bochow, B., Kellum, C., 2010. Vehicular Networking: Automotive Applications and Beyond. John Wiley & Sons

Yu, C., Feng, Y., Liu, H.X., Ma, W. and Yang X., 2018. Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections. *Transportation Research Part B: Methodological,* 112, pp. 89-112

Marler, R.T., Arora, J.S., 2004. Survey of multi-objective optimization methods for engineering. Struct. Multidiscipl. Optim. 26, 369–395.

Jiang, H., Hu, J., An, S., Wang, M., Park, B.B., 2017. Eco approaching at an isolated signalized intersection under partially connected and automated vehicles environment. Transp. Res. Part C 79, 290–307.

Frey, H., Unal, A., Chen, J., Li, S., Xuan, C., 2002. Methodology For Developing Modal Emission Rates For EPA's Multi-Scale Motor Vehicle & Equipment Emission System. Ann Arbor, Michigan: US Environmental Protection Agency. North Carolina State University for Office of Transportation and Air Quality, U.S. Environmental Protection Agency, Ann Arbor, MI.

Gurobi Optimization Inc., 2017. Gurobi Optimizer Reference Manual.

Bilmes, J.A. et al, 1998. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models.
Int. Comput. Sci. Inst. 4 (510), 126.

Zheng, J. and Liu, H.X., 2017. Estimating traffic volumes for signalized intersections using connected vehicle data. *Transportation Research Part C: Emerging Technologies*, *79*, pp.347-362.

Day, C. M., and D. M. Bullock. Detector-Free Signal Offset Optimization with Limited Connected Vehicle Market Penetration. Transportation Research Record: Journal of the Transportation Research Board, Vol. 2558, 2016, pp. 54–65. https://doi.org/10.3141/2558-06.

Feng, Y., Zheng, J. and Liu, H.X., 2018. Real-Time Detector-Free Adaptive Signal Control with Low Penetration of Connected Vehicles. *Transportation Research Record*, Vol. 2672(18) 35-44

Zheng, J., Sun, W., Huang, S., Shen, S., Yu, C., Zhu, J., Liu, B., and Liu, H.X., 2019. Traffic Signal Optimization Using Crowdsourced Vehicle Trajectory Data. *Transportation Research Part C: Emerging Technologies (Under Review)*

Project Output:

A. Publications:

The team has presented in the DOE annual merit review meeting in June 2018.
Twelve papers have been submitted by the team for publications:

1. Di, X., Zhao, Y., Zhang, Z., and Liu, H., Data-Driven Similarity Analysis for Activity-based Travel Demand Modeling, Submitted to 2017 TRB Annual Meeting for Presentation.
2. Huang, X. and Peng, H., 2017, May. Speed trajectory planning at signalized intersections using sequential convex optimization. In American Control Conference (ACC), 2017(pp. 2992-2997). IEEE.
3. Di, X., Liu, H.X., Ban, X., Yang, H., 2016. Ridesharing User Equilibrium and Its Implications for High-Occupancy Toll Lane Pricing, Transportation Research Record, 2667, doi: 10.3141/2667-05.
4. Huang, X. & Peng, H., 2018. Eco-Routing based on a Data Driven Fuel Consumption Model. To appear in 14th International Symposium on Advanced Vehicle Control, AVEC'18.
5. Huang, X. & Peng, H., 2018. Efficient Mobility-on-Demand System with Ride-Sharing. Submitted to The 21st IEEE International Conference on Intelligent Transportation Systems.
6. Zheng, J. and Liu, H.X., 2017. Estimating traffic volumes for signalized intersections using connected vehicle data. Transportation Research Part C: Emerging Technologies, 79, pp.347-362.
7. Feng, Y., Yu, C. and Liu, H.X., 2018. Spatiotemporal Intersection Control in a Connected and Automated Vehicle Environment. Transportation Research Part C: Emerging Technologies, 89, pp.364-383.
8. Yu, C., Feng, Y., Liu, H.X., Ma, W. and Yang X., 2018. Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections. Transportation Research Part B: Methodological.
9. Feng, Y., Zheng, J. and Liu, H.X., 2018. A Real-time Detector-Free Adaptive Signal Control with Low Penetration of Connected Vehicles. Transportation Research Record. (In Press)
10. Bo Yu, Shan Bao, Fred Feng, James R Sayer (2019). Examination and Prediction of Drivers' Reaction when Provided with V2I Communication-based Intersection Maneuver Strategies, Transportation Research Part C: Emerging Technologies, 106, 17-28.

11. Zhao, Y., Guo, X. and Liu, H.X., 2019. The impact of autonomous vehicles on commute ridesharing with uncertain work end time. Submitted to Transportation Research Part B: Methodological.
12. Oh, G. and Peng, H., "Eco-driving at signalized intersections: What is possible in the real-world?" 2018 21st International Conference on Intelligent Transportation Systems (ITSC) (pp. 3674-3679).

B. Technologies/Techniques:
This project produced several key outcomes, as highlighted below.

- Collected energy consumption + GPS data from ~500 vehicles, which is from 8 million miles of naturalistic driving.
- A calibrated Ann Arbor model in Polaris (ANL), Based on detailed analysis, the fuel economy prediction accuracy is at about 3.9%.
- An open-source SUMO model of Ann Arbor was also developed. This is done partially by one of the graduate student who worked on this project, as well as another student supported by an ARPA-E NEXTCAR project.
- Eco-Routing in Ann Arbor (SUMO model) shows 6% fuel saving potential.
- Human behavior model (72% follows Eco-driving suggestions, each shared vehicle can replace 4 individually owned vehicles).
- Human driver etiquette based on human vehicle data (how people drive), from 13 driving behavior data analysis.
- Adaptive Traffic Signal Control Algorithm was developed based on the data collected, and traffic network of the City of Ann Arbor (For CAVs, 13% delay reduction, 10% fuel reduction).

C. Status Reports:
This is the final report of the project.

D. Media Reports:
None

E. Invention Disclosures:
None

F. Patent Applications:
None

G. Licensed Technologies:
None

H. Networks/Collaborations Fostered:
The team has worked closely with EPA, and with significant data available from the fleet (8 million miles), test data have been shared with ANL, INL and a smaller subset of de-identified data have been shared with EPA and UM student teams working in the TechLab program (which pairs students with startup companies). Selected results were presented to the Mcity

leadership Circle companies in February, 2018. Ford Motor Company expressed interest in the progress of this project and the UM research team visited Ford Research and Innovation Center at Dearborn to provide a presentation. The PI of this project was invited to give two high-level presentations: the ORNL organized Smoky Mountain Mobility Summit (Oct. 3&4), and the National Academy of Science organized Light-duty vehicle technology review (Oct 15). Key findings from this project were highlighted.

I. Websites Featuring Project Work or Results:
None

J. Other Products:
None

K. Awards, Prizes, and Recognition:
None