

12-2013

# Enhancing Energy Efficiency in Connected Vehicles Via Access to Traffic Signal Information

Grant Mahler  
*Clemson University*

Follow this and additional works at: [https://tigerprints.clemson.edu/all\\_dissertations](https://tigerprints.clemson.edu/all_dissertations)



Part of the [Mechanical Engineering Commons](#)

---

## Recommended Citation

Mahler, Grant, "Enhancing Energy Efficiency in Connected Vehicles Via Access to Traffic Signal Information" (2013). *All Dissertations*. 1612.

[https://tigerprints.clemson.edu/all\\_dissertations/1612](https://tigerprints.clemson.edu/all_dissertations/1612)

This Dissertation is brought to you for free and open access by the Dissertations at TigerPrints. It has been accepted for inclusion in All Dissertations by an authorized administrator of TigerPrints. For more information, please contact [kokeefe@clemson.edu](mailto:kokeefe@clemson.edu).

ENHANCING ENERGY EFFICIENCY  
IN CONNECTED VEHICLES  
VIA ACCESS TO  
TRAFFIC SIGNAL INFORMATION

---

A Doctoral Dissertation  
Presented to  
The Graduate School of  
Clemson University

---

In Partial Fulfillment of  
the Requirements for the Degree  
Doctor of Philosophy  
Mechanical Engineering

---

by  
Grant Mahler  
December 2013

---

Presented to:  
Dr. Ardalan Vahidi, Committee Chair  
Dr. Mohammed Daqaq  
Dr. Thomas Kurfess  
Dr. Joshua Summers  
Dr. Paul Venhovens



# Abstract

This dissertation expounds on algorithms that can deterministically or probabilistically predict the future Signal Phase and Timing (SPAT) of a traffic signal by relying on real-time information from numerous vehicles and traffic infrastructure, historical data, and the computational power of a back-end computing cluster. When made available on an open server, predictive information about traffic signals' states can be extremely valuable in enabling new fuel efficiency and safety functionalities in connected vehicles: Predictive Cruise Control (PCC) can use the predicted timing plan to calculate globally optimal velocity trajectories that reduce idling time at red signals and therefore improve fuel efficiency and reduce emissions. Advanced engine management strategies can shut down the engine in anticipation of a long idling interval at red. Intersection collision avoidance is another functionality that can benefit from the prediction.

We start by exploring a globally optimal velocity planning algorithm through the use of Dynamic Programming (DP), and provide to it three levels of traffic signal information - none, real-time only, and full-future information. The no-information case represents the average driver today, and is expected to provide an energy efficiency minimum or baseline. The full-information case represents a driver with full

and exact knowledge of the future red and green times of all the traffic signals along their route, and is expected to provide an energy efficiency maximum. We propose a probabilistic method that seeks to optimize fuel efficiency when only real-time only information is available with the goal of obtaining fuel efficiency as close to the full-future knowledge example as possible. We used Monte-Carlo simulations to evaluate whether the fuel efficiency gains found were merely the result of lucky case studies or whether they were statistically significant; we found in related case studies that up to 16% gains in fuel economy were possible. While these results were promising, the delivery of relevant and accurate future traffic signal phase and timing information remained an unsolved problem.

The next step we took was towards building traffic signal prediction models. We took several prescient techniques from the data mining and machine learning fields, and adapted them to our purposes in the exploration of massive amounts of data recorded from Traffic Management Centers (TMCs). This manuscript evaluates Transition Probability Modeling, Decision Tree, Multi-Linear Regression, and Neural Network machine learning methods for use in the prediction of traffic Signal Phase and Timing (SPaT) information.

Finally, we evaluated the influence of providing SPaT data to vehicles. To that end, we investigated both smartphone and in-vehicle proof-of-concepts. An in-vehicle velocity recommendation application has been tested in two cities: San Jose, California and San Francisco, California. The two test locations used two different data sources: data directly from a TMC, and data crowdsourced from public transit bus routes, respectively. A total of 14 test drivers were used to evaluate

the effectiveness of the algorithm. In San Jose, the algorithm was found to produce a 8.4% improvement in fuel economy. In San Francisco, traffic conditions were not conducive to testing as the driver was unable to significantly vary his speed to follow the recommendation algorithm, and a negligible difference in fuel economy was observed. However, it did provide an opportunity to evaluate the quality of data coming from the crowdsourced data algorithms. Predicted phase timing compared to camera-recorded ground truth data indicated an RMS difference (error) in prediction of approximately 4.1 seconds.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Research Overview . . . . .	4
<b>2</b>	<b>Velocity Recommendation Algorithm using Traffic Signal Phase and Timing Data</b>	<b>7</b>
2.1	Abstract . . . . .	7
2.2	Introduction . . . . .	8
2.3	Optimal Velocity Planning . . . . .	12
2.3.1	Planning with Deterministic SPAT Information . . . . .	13
2.3.2	Planning with Probabilistic SPAT Information . . . . .	15
2.4	Prediction . . . . .	16
2.5	Simulations . . . . .	20
2.5.1	Motivating Demonstration . . . . .	22
2.5.2	Monte-Carlo Simulations . . . . .	25
2.5.3	Simulations Using Recorded Timings from Arterial Adaptive Lights . . . . .	29
2.6	Conclusions . . . . .	34
2.7	Acknowledgement . . . . .	35
<b>3</b>	<b>Machine Learning Prediction of Actuated Traffic Signal Phase and Timing Data</b>	<b>36</b>
3.1	Abstract . . . . .	36
3.2	Introduction . . . . .	37
3.3	Background on Traffic Signal Control . . . . .	40
3.4	Raw Data Acquisition . . . . .	44
3.4.1	Raw Data (Attributes) . . . . .	45
3.5	Data Preparation and Exploration . . . . .	48

3.5.1	Principal Component Analysis . . . . .	48
3.5.2	Variance of Phases . . . . .	50
3.6	Prediction of Pre-Timed Signals . . . . .	51
3.7	Prediction of Actuated and Adaptive Signals . . . . .	54
3.7.1	Transition Probability Modeling Approach . . . . .	56
3.7.2	Decision Tree . . . . .	57
3.7.3	Multi-Linear Regression . . . . .	58
3.7.4	Neural Networks . . . . .	59
3.8	Discussion . . . . .	60
3.8.1	Results . . . . .	60
3.8.2	Sources of Error . . . . .	64
3.8.3	Discussion . . . . .	68
3.9	Acknowledgement . . . . .	68
<b>4</b>	<b>In-Vehicle Velocity Recommendation Architecture Evaluation</b>	<b>69</b>
4.1	Disclosure . . . . .	69
4.2	Abstract . . . . .	70
4.3	Introduction . . . . .	70
4.4	System Architecture . . . . .	73
4.4.1	Vehicle Subscriptions . . . . .	74
4.4.2	System Backend . . . . .	76
4.5	Data Sources . . . . .	76
4.5.1	Traffic Management Center . . . . .	77
4.5.2	Crowdsourced Data via GPS Traces . . . . .	78
4.5.3	Crowdsourced Data via On-Board Cameras . . . . .	79
4.6	Speed Recommendation as Exemplary Use Case . . . . .	80
4.6.1	Velocity Advisory Algorithm . . . . .	81
4.6.2	Fuel Economy Evaluations . . . . .	86
4.7	iPhone Implementation . . . . .	92
4.8	Problems Encountered . . . . .	94
4.9	Conclusions . . . . .	97
4.10	Acknowledgement . . . . .	98
<b>5</b>	<b>Summary of Contributions and Proposed Future Work</b>	<b>99</b>
5.1	Review of Results . . . . .	99
5.2	Novel Contributions . . . . .	102
5.3	Takeaways . . . . .	104
5.4	Dissemination of Results . . . . .	106



5.5	Future Work . . . . .	107
<b>A</b>	<b>Appendices</b>	<b>116</b>
A.1	Supplemental Experimental Setup Information . . . . .	116

# List of Figures

1.1	A vehicle with sufficient information is able to avoid red lights [1]. . .	2
2.1	Schematic of velocity planning based on probability of green for two consecutive lights . . . . .	10
2.2	Conditional future probability of green given that the light is currently green, for four different light timing patterns. In all patterns the total cycle time is 60 seconds, with the lengths of green and red indicated in the legends. The time axis is $t_p$ as described in Equations 2.5 and 2.6. . . . .	18
2.3	Conditional future probability of green given that the light is currently red, for four different light timing patterns. In all patterns the total cycle time is 60 seconds, with the lengths of green and red indicated in the legends. The time axis is $t_p$ as described in Equations 2.5 and 2.6. . . . .	19
2.4	Case Study: Comparison of velocity profiles of a driver with no information, real-time information, and full horizon information. . . .	23
2.5	Snapshots of a running simulation for Case Study 1. Probability prediction is indicated by the red to green color spectrum. See the video at the following link [2]. . . . .	24
2.6	Average fuel economy for increasing number of simulated streets. . . .	27
2.7	Histogram of Monte-Carlo simulation results for the three levels of information. . . . .	28
2.8	Histories of relevant phases for each light along the chosen real-world route, for every cycle over a 24 hour period (midnight to midnight). .	30
2.9	Histogram of fuel economies for real-world traffic signal timings. . . .	32
3.1	Phase numbering of a standard intersection per FHWA standards [3].	41
3.2	Overview of signal control schemes in place around the United States.	41
3.3	A standard (leading left) ring-barrier diagram [4]. . . . .	42

3.4	Twenty four hours of training data for an example signal at each of the three test locations. Graphs show only data from Ring 1, and have been abbreviated in the Y-axis, cutting off outliers in order to show the phase length variance in the majority of cycles. . . . .	46
3.5	PCA results. . . . .	49
3.6	Conditional probability of green given green now, for four different light timing patterns. In all patterns the total cycle time is 60 seconds, with the lengths of green and red indicated in the legends. The time axis is $t_p$ as in Equations 3.1 and 3.2. . . . .	52
3.7	Conditional probability of green given red now, for four different light timing patterns. In all patterns the total cycle time is 60 seconds, with the lengths of green and red indicated in the legends. The time axis is $t_p$ as in Equations 3.1 and 3.2. . . . .	53
3.8	Inputs and outputs of machine learning prediction models. . . . .	55
3.9	Transition probability for Phase 4 of Signal 0001 in Portland, Oregon based on data collected 09/24/2012. . . . .	57
3.10	Method performance given varying amounts of data (from signal 0035 in Fremont, CA). Hours of data (for training, also for testing) given on X-axis, mean performance across all 8 phases given across Y-axis. . . . .	60
3.11	Root mean square error divided by mean phase length for test data from Fremont California, Portland Oregon, and San Jose California, with algorithms provided 19 attributes. . . . .	62
4.1	System overview of the path of data from a traffic signal to server to a vehicle. The path of data from the vehicle camera is described in more detail in Section 4.5 Subsection 4.5.3. . . . .	74
4.2	Driver's dash display, including speed recommendation and count-down to green. . . . .	80
4.3	Schematics map of red lights distributed over space-time. The graphics shows how an informed vehicle passes two consecutive traffic intersection without having to stop at a red. . . . .	84
4.4	Fuel economy evaluation route in San Jose, California. . . . .	88
4.5	Fuel economy evaluation route in San Francisco, California. . . . .	90
4.6	Screenshots of our iPhone application that uses SPAT information to calculate the speed to reduce idling at reds. On the left: dynamically changing optimal speed. On the right: instantaneous state of upcoming lights on Pleasantburg Drive in Greenville, SC. . . . .	93

# List of Tables

2.1	Fuel economies in miles per gallon (mpg). The results show the impact information can have on the energy efficiency. . . . .	24
2.2	Monte-Carlo simulation results reflect the positive influence of information, on average, on fuel economy. . . . .	26
2.3	Fuel economy results from recorded real-world traffic signal timings with simulated vehicles moving between the lights reflect the positive influence of information. . . . .	32
3.1	Standard Deviation (in seconds) of Training Data . . . . .	50
3.2	Neural Networking neurons-per-layer sensitivity. All units are seconds.	59
4.1	Field testing in San Jose, California. Drivers were <i>not</i> given specific instructions to follow speed recommendations. . . . .	88
4.2	Field testing in San Jose, California. Drivers <i>were</i> given specific instructions to follow speed recommendations. . . . .	89
4.3	Field testing in San Francisco, California. Fuel economies in miles per gallon (mpg). . . . .	90

# Chapter 1

## Introduction

### 1.1 Motivation

National security, economic, and environmental policies all indicate that many countries could benefit from reduced oil consumption. Transportation is a significant consumer of oil, and while electric and hydrogen vehicles offer promise, the required infrastructure changes would be expensive. This yields a technological niche which is ripe for development, where techniques to improve existing internal combustion engine, and hybrid-electric vehicles are sought. With this in mind, the recent development of sensors on- and off-board vehicles has led to a boon of information. Simultaneously, communication bandwidth, data storage, and computational power have become relatively cheap [5]. It is at this junction that opportunity exists to reduce oil consumption by improving vehicle fuel efficiency. Vehicles idling at red lights cause emissions and noise pollution, and time spent idling can reduce a vehicle's average fuel economy. Optimizing signal timings and use of more advanced traffic signal control software and hardware can help reduce idling times but is quite costly

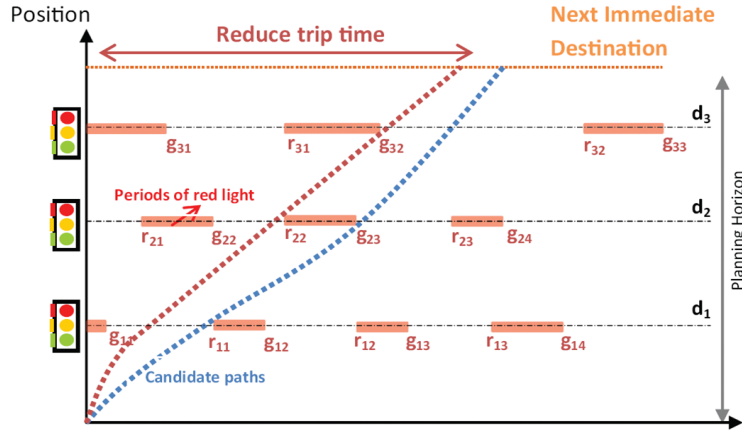


Figure 1.1: A vehicle with sufficient information is able to avoid red lights [1].

[6, 7, 8, 9]. Very recent research on developing velocity planning or advisory algorithms based on Signal Phase and Timing (SPAT) information [10, 11, 12, 13, 14, 15] has the potential to reduce inefficiencies caused by traffic signals, without significant changes to public infrastructure or automotive production paradigms.

In [11], Vahidi's group presented a velocity planning algorithm which guided a driver through multiple traffic signals given full and exact knowledge of the future state of traffic signals. Figure 1.1 is an intuitive way of understanding the proposed approach. Provided prior knowledge of SPAT information, each vehicle can potentially plan a velocity that reduces its idling time behind red lights. Simulation results using the algorithm in [11] indicated approximately 24% increase in fuel economy when passing through a series of simulated traffic signals. Koukoumidis et al. have recently developed an iPhone application to guide a driver through a light [13]. An underlying assumption in both papers is that accurate SPAT information is readily

available. Unfortunately due to timing drift in fixed-time lights, and ever-changing traffic conditions for actuated and adaptive lights, accurate SPAT information is difficult to obtain.

In [13], a machine learning technique, Support Vector Machine (SVM), was used to predict the phase of the light based on information from windscreen mounted iPhone cameras. Each iPhone utilized an image analysis algorithm to determine the current phase of the signal, and generated an ad-hoc wireless network to distribute this information to other iPhones running the same application. While SVM is one potential method of prediction, the application's reliance on ad-hoc wireless networks and image processing to provide signal timing has shortcomings. In [12], an iPhone application to re-route drivers around red lights was presented. Unfortunately the prediction method was not explicitly indicated.

The proposed research in this dissertation is aimed at addressing these shortcomings to enable widespread adoption of technologies that rely on prediction of SPAT information. We start with a statistical evaluation of the effects on fuel economy of a dynamic programming speed recommendation algorithm. We then focus on SPaT prediction using various data sources and data analysis techniques. Finally, in-vehicle velocity advisory tests were performed to validate effects of providing SPaT information to in-vehicle algorithms.

## 1.2 Research Overview

In Chapter 2, a method is proposed which utilizes either the base timing plan or historically averaged timing data, in complement with real-time signal phase data, to produce probabilistic SPAT prediction. This SPAT prediction will be fed into a Dynamic-Programming algorithm in order to empirically examine whether the prediction accuracy is sufficient to improve the fuel economy results. As a result of its reliance only on real-time data instead of full-horizon data, this potentially overcomes issues of timing drift and unknown traffic conditions at actuated and adaptive signals. While the results are promising (and can be found in Table 2.1), the first models are only able to recover about half of the potential fuel savings of a vehicle with full and exact deterministic knowledge of future traffic signal phase and timing information. The traffic signal prediction aware velocity recommendation from the DP is initially evaluated with respect to simpler fixed-time signals. A Monte-Carlo simulation is used to evaluate the performance in light of a multitude of different traffic signal phase and timing conditions. The probabilistic prediction DP velocity recommendation algorithms is then evaluated with actuated and adaptive traffic signals, using traffic signal phase and timing information recorded over a 24 hour period from a street in Fremont, California. The results indicate that i. knowledge of traffic signal timing can have a significant effect on fuel efficiency and ii. future knowledge of traffic signal timing, and evaluation of the same, in real world situations are important next steps.

To recover more of the potential fuel savings of full future SPAT information,



and to expand the applicability to all types of traffic signals, in Chapter 3 data mining and machine learning methods for producing future SPAT information will be explored. Methods pulled from the computer science field, including Transition Probability Modeling (TPM), Decision Trees (DT), Multi-Linear Regression (MLR), and Neural Networks(NN) will be compared. The goal is prediction accuracy, measured as root mean square error between recorded ground truth data and predicted phase length data, using only historical and real-time information directly available from traffic signals. Final models used the attributes elapsed phase length, a three cycle mean, a ten cycle mean, the time vehicles have been waiting at inactive phases for all 8 phases, and the vehicle call status for all 8 phases, in both training and testing the algorithms. Due to the diverse age and brands of technologies implemented in traffic signal controllers, and the non-deterministic influence of vehicle and pedestrian arrival, prediction of the next active phase(s) and phase length is a challenging problem. Several of the methods prove promising enough for consideration as part of in-vehicle algorithms, potentially including automatic motor start-stop and gear selection.

With accurate signal phase and timing information from Chapter 3 in hand, in Chapter 4 opportunities for use of this information will be explored. Both smartphone based and on-board vehicle solutions are feasible. Chapter 4 focuses primarily on in-vehicle testing during the time period at the BMW Technology Office USA and University of California at Berkeley, where a development 2011 535i was modified in order to implement and evaluate the effectiveness of the project as a whole. A total of 14 drivers evaluated an in-vehicle velocity recommendation system, to positive

effect. Testing methodology, including driver preparation, is surprisingly important to test results. Testing was performed in both San Jose, California and San Francisco, California as evaluations of traffic management center and crowdsourced data sources, respectively. A proof of concept iPhone-based velocity recommendation application which was tested in Greenville, South Carolina is also presented.

Chapter 5 provides an overall discussion of results and conclusions from previous chapters, describes the novel contributions, provides a record of dissemination of results, and explores potential directions for future work.

# Chapter 2

## Velocity Recommendation

## Algorithm using Traffic Signal

## Phase and Timing Data

### 2.1 Abstract

The main contribution of this chapter is the formulation of a predictive optimal velocity planning algorithm that uses probabilistic traffic Signal Phase And Timing (SPAT) information to increase a vehicle's energy efficiency. We introduce a signal phase prediction model which uses historically-averaged timing data and real-time phase data to determine the probability of green for upcoming traffic lights. In an optimal control framework, we then calculate the best velocity trajectory that maximizes the chance of going through greens. Case study results from a multi-

signal simulation indicate that energy efficiency can be increased with probabilistic timing data and real-time phase data. Monte-Carlo simulations are used to confirm that the case study results are valid, on average. Finally, simulated vehicles are driven through a series of traffic signals, using recorded data from a real-world set of traffic-adaptive signals, to determine the applicability of these predictive models to various types of traffic signals.

## 2.2 Introduction

A significant amount of fuel is spent by vehicles slowing down, sitting behind, and accelerating away from traffic signals [8, 9, 16]. With Corporate Average Fuel Economy standards set to rise, new technologies must be developed to meet the more stringent standards. Avoidance of red signals could improve vehicle specific fuel economy, reduce emissions, and help automotive manufacturers meet these new standards. While cities can spend time and money improving the timing of traffic signals [7], new research in velocity advisory algorithms suggests that it is possible to avoid red traffic signals through intelligent usage of traffic signal phase and timing information [11, 12, 13, 14, 15, 17]. This benefit comes without the cost imposed by significant changes to infrastructure or production vehicles.

A velocity planning algorithm which guided a driver through multiple traffic signals given full and exact knowledge of the future state of traffic signals has previously been published by our group [11]. This algorithm could be implemented as a smart phone application displaying a suggested velocity to the driver, or it could

provide the reference velocity to the adaptive cruise control system of a car. Simulation results using the algorithm in [11] indicated approximately 24% increase in fuel economy when passing through a series of simulated traffic signals. Koukoumidis et al. have presented an iPhone application to guide a driver through a light [13]. An underlying assumption in both papers is that accurate Signal Phase And Timing (SPAT) information is readily available. Unfortunately due to timing drift in fixed-time lights, and ever-changing traffic conditions for actuated and adaptive lights, accurate SPAT information is difficult to obtain.

In this chapter we propose a method which utilizes either the base timing plan or historically averaged timing data (for example a 24 hour average), in complement with real-time signal phase data, to produce probabilistic SPAT predictions. A velocity planning algorithm then uses the prediction to reduce the chance of idling at a red light. A schematic is shown in Figure 2.1. This method can be implemented today as a result of its reliance only on real-time data instead of full-horizon data, and potentially overcomes issues of timing drift and unknown traffic conditions at actuated and adaptive signals.

In [13], a machine learning technique, Support Vector Machine (SVM), was used to predict the current phase of the light based on information from windscreen mounted iPhone cameras. Each iPhone utilized an image analysis algorithm to determine the current phase of the signal, and generated an ad-hoc wireless network to distribute this information to other iPhones running the same application. While SVM is one potential method of prediction, the application's reliance on ad-hoc wireless networks and image processing to provide signal timing has shortcomings.

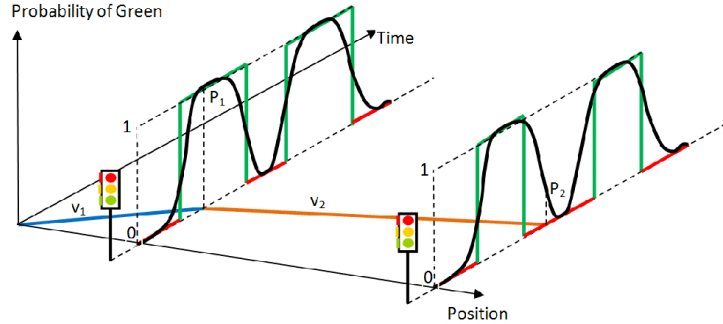


Figure 2.1: Schematic of velocity planning based on probability of green for two consecutive lights

In [4], several traffic signal phase length prediction algorithms were presented, with the goal of utilizing the information in vehicle efficiency applications. However, the accuracy of the proposed methods as phase length predictor algorithms was deemed insufficient by the thesis author. In [18], the authors looked at clustering of velocity profiles from drivers who previously visited a road segment to determine traffic signal state estimations; but the authors do not make phase length predictions. In [12], an iPhone application to re-route drivers around red lights was presented. Unfortunately the prediction method was not explicitly indicated. In current literature, there exists no comprehensive framework for velocity planning when inexact or incomplete traffic signal phase and timing information is available.

The goal of this chapter is to fill the gap in current research, by developing SPAT prediction and probabilistic velocity planning algorithms that are applicable to both fixed time and actuated signals, in the presence of both exact and inexact red/green split information. An optimal control formulation for the velocity

planning problem is presented in Section 2.3. Section 2.4 describes a procedure for predicting, probabilistically, the future phase of a signal, based on its current phase and averaged timing data. A full array of simulations in Section 2.5 expands on our initial results in [19] and statistically evaluates fuel economy gains attainable with our proposed methods versus cases with no signal information (baseline minimum efficiency) and full horizon signal information (maximum attainable efficiency). In particular in Section 2.5.1, we present a motivating simulation case study consisting of three consecutive traffic signals with our selection of phase and timing configurations. Next in Section 2.5.2 we move to a simulation containing a large number of test cases, involving randomized traffic signal timings of known red/green split; this builds statistically significant evidence for use when the driver has access to exact red/green split information akin to a driver passing through a series of fixed time lights. To further demonstrate the applicability of proposed methods, in Section 2.5.3 we show results for a series of consecutive traffic signal phase and timing configurations matching a series of real-world intersections. In these simulations we utilize actuated traffic signal timings, pre-recorded from an actual street. The goal of using the signal phase and timings from the real world is two-fold: first it allows the evaluation of the algorithm in the presence of actuated signals and inexact red/green splits, and secondly it reduces author influence on the road geometry and signal timings.

## 2.3 Optimal Velocity Planning

Our goal is to find a velocity profile which reduces the total energy consumption during a trip based on full or partial SPAT information. One can formulate this problem as an energy (fuel) minimization problem, but this requires inclusion of dynamic models of a specific vehicle and its propulsion system (combustion engine, etc.) to relate energy use to the velocity profile (for example, see Rakha et al.[20]). To avoid the ensuing computational complexity and to decouple the choice of optimal speed from a vehicle’s make and model, we simplify the cost function by penalizing in it a weighted sum of the total trip time and all of the acceleration and decelerations, instead of total energy use. The underlying assumptions in this choice are that idling at a traffic light and excessive accelerations and decelerations induced by a traffic light cost energy with no benefits to the driver. Other factors such as motion constraints imposed by red intervals, road speed limits, and the fact that very low velocities will be unacceptable to consumers, can be accounted for by constraining the solution space. We evaluate the fuel economy for a specific vehicle model a-posteriori, by feeding the optimal velocity profile to a high-fidelity dynamic model of the vehicle.

We first describe, in Section 2.3.1, the scenario when deterministic and accurate SPAT information over the entire planning horizon is available. When the phase and timing of upcoming signals are uncertain, a probabilistic term can be added to the cost function, as described in Section 2.3.2.



### 2.3.1 Planning with Deterministic SPAT Information

To obtain a best achievable energy efficiency baseline, we first solve the optimal control problem assuming full and deterministic knowledge of signals' phase and timing over the planning horizon. The following cost function is used:

$$J = \sum_i \left[ w_1 \frac{t_{i+1} - t_i}{\Delta t_{min}} + w_2 \left| \frac{a_i}{a_{max}} \right| + c(x_i, t_i) \frac{1}{\epsilon} \right] \quad (2.1)$$

where  $J$  is the total cost and is indexed over position  $x$  with index  $i$ ,  $t_{i+1} - t_i$  is the time required for a vehicle to cover the distance between steps  $x_i$  and  $x_{i+1}$  given the velocity at  $x_i$  and the acceleration  $a_i$ ;  $\Delta t_{min}$  is the minimum time to complete the step if starting and ending at the maximum velocity and is used as a scaling factor,  $a_i$  is the constant acceleration assumed during step  $i$ , and  $a_{max}$  is the maximum allowed acceleration. The constants  $w_1$  and  $w_2$  are weighting terms. Motion constraints imposed by a red interval are imposed as a soft constraint by inclusion of the term  $c(x_i, t_i) \frac{1}{\epsilon}$  in the cost function<sup>1</sup>. The value of  $c(x_i, t_i)$  is zero except for spatiotemporal intervals when a light is red in which case its value is set to one, and  $\epsilon$  is a very small constant (for example  $10^{-6}$ ), such that idling at red is discouraged.

The vehicle kinematics, realized by the following two-state equations, are imposed as equality constraints. Here  $x$  is the independent variable, velocity  $v$  and time  $t$  are the two states, and acceleration  $a$  is the input:

---

<sup>1</sup>Note that in simulations, a low level controller verifies and can override the recommendation of the velocity planner, if the planner makes a recommendation which would pass through a red light.

$$\begin{cases} \frac{dv}{dx} = \frac{a}{v} \\ \frac{dt}{dx} = \frac{1}{v} \end{cases} \quad (2.2)$$

Discretizing the above equations with a constant sampling interval of  $\Delta x = x_{i+1} - x_i$  and with a zero-order hold on acceleration, we obtain:

$$\begin{cases} v_{i+1} = \sqrt{(v_i)^2 + 2a_i\Delta x} \\ t_{i+1} = t_i + \frac{2\Delta x}{v_i + \sqrt{(v_i)^2 + 2a_i\Delta x}} \end{cases} \quad (2.3)$$

We also enforce the hard inequality constraints:  $v_{min} \leq v_i \leq v_{max}$  and  $a_{min} \leq a_i \leq a_{max}$ . Here  $v_{min}$  and  $v_{max}$  are the road speed limits and can also include lowest speed acceptable to a driver;  $-a_{max}$  and  $a_{max}$  are the feasible bounds for deceleration and acceleration.

The above optimal control problem is solved numerically using Deterministic Dynamic Programming (DDP) and based on the discretization on position, time, and velocity[21]. The DDP is solved by value function iterations for each stage, backwards. Using Bellman's principle of optimality, one only has to solve for one control input, here  $v_i$ . The trajectory is found recursively, instead of attempting to find the whole velocity trajectory at once.

### 2.3.2 Planning with Probabilistic SPAT Information

Because perfect full-horizon SPAT information is generally not available, a solution which takes advantage of currently available information and technologies is preferable. The goal is a solution that, given imperfect or incomplete phase and timing information, is still able to increase the energy efficiency by taking advantage of available data. Because of imperfect or incomplete starting data, this resulting energy efficiency is expected to be lower than the case with full-horizon information.

The cost function in (2.1) is modified to the following to take into account the probabilistic nature of SPAT information:

$$J = \sum_i \left[ w_1 \frac{t_{i+1} - t_i}{\Delta t_{min}} + w_2 \left| \frac{a_i}{a_{max}} \right| + c(x_i, t_i) |\log_e(p(x_i, t_i))| \right] \quad (2.4)$$

All parameters and variables in (2.4) are the same as those described for (2.1); the only new variable is  $p(x_i, t_i)$  which represents probability of green at time  $t_i$  for a light situated at position  $x_i$ . Therefore higher costs are assigned to solutions that pass through time intervals where probability of green is lower. At the limit when probability of green at  $x_i, t_i$  is zero,  $\log_e(p(x_i, t_i)) = \infty$  and passing through a red would be discarded. Where  $p(x_i, t_i) = 1$ , this term of the cost function drops to zero and increases the likelihood that the corresponding velocity will be selected. The probability of green for each light can be generated based on real-time and/or historical information as described in Section 2.4. Minimization of the cost function (2.4) with the equality and inequality constraints described in the previous subsection, remains a deterministic optimal control problem. The problem

is solved using DDP but in a receding horizon manner; as new information becomes available, the DDP is re-solved taking into account the updated information over the remaining trip horizon.

## 2.4 Prediction

There can be much uncertainty in the phase and timing of a traffic signal which makes predicting its future state quite challenging. For fixed-time traffic signals which do not respond to traffic conditions and operate only on a timing table, we have confirmed the finding that the traffic signal clock drifts significantly during a 24 hour period. Therefore, it is not possible to know with certainty the start of greens and reds, even for fixed-time signals. The level of uncertainty is higher for actuated and adaptive traffic signals which do respond to traffic conditions. Although they have a base timing table, the timings of actuated and adaptive lights may change according to traffic conditions, rendering not only the start of reds and greens but also the phase lengths uncertain.

Due to the aforementioned uncertainties, it is difficult to determine the start and duration of greens deterministically. Therefore in this paper we employ a probabilistic prediction framework to handle the case with partial or uncertain information. We focus on cases where only i) the current phase (color) and ii) the average red and green lengths for a signal are known. We use this information to predict the probability of a green over the planning horizon.

Access to the current phase of the traffic signal is a major technological hurdle.

However, solutions have been proposed and implemented in [12, 22] that could address this problem. Other approaches, including those that rely on Dedicated Short Range Communication (DSRC), can be found in [11, 23, 24].

Obtaining a base timing plan for a traffic signal is not trivial either. Direct access to signal timing plans is prohibitively difficult due to hundreds of local and federal entities that manage the more than 330,000 traffic lights across the United States [7]. To overcome these problems, it is possible to combine historical data, operating logic of signalized intersections, infrastructure sensor data, and even crowd source information to generate an average timing table. This can be done for different times in a day (rush hour/midday) and days of a week (weekday/weekend). The outcomes are average cycle times, and percentage of green and red in each travelling direction for each signal. Mere knowledge of such a baseline schedule, obtained offline and using only historical data, has statistical value even when the signal clock time is unknown.

Let us denote the state of a light by  $\ell(t)$  which can assume two values,  $g$  and  $r$ , representing green and red respectively. We are interested in determining the probability of a light being green at time  $t + t_p$  conditioned on its current color at time  $t$ . To form this conditional probability function, we assume the durations of green and red are known to be  $t_g$  and  $t_r$  on average. We also assume the traffic signal operates cyclically<sup>2</sup> and as a result the total cycle time<sup>3</sup> is fixed and equal to

---

<sup>2</sup>This is true for many traffic signals; even many of those that react to traffic can theoretically have a fixed cycle time.

<sup>3</sup>We include the yellow time with red time; for safety reasons we do not make recommendations which would guide a driver through a yellow light.

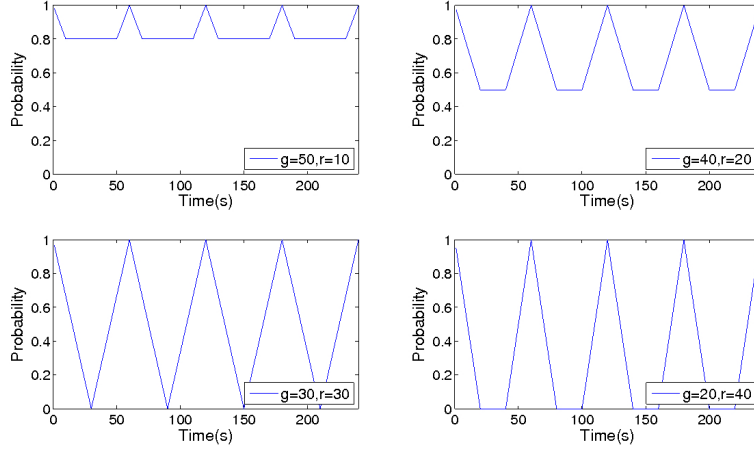


Figure 2.2: Conditional future probability of green given that the light is currently green, for four different light timing patterns. In all patterns the total cycle time is 60 seconds, with the lengths of green and red indicated in the legends. The time axis is  $t_p$  as described in Equations 2.5 and 2.6.

$t_g + t_r$ . In this formulation, we assume the arrival of vehicles at the intersection to be uniformly distributed; if the arrival distribution of vehicles at an intersection is known (for example, in [25]), that distribution may be used as a weighting function in place of the uniform assumption. Using relatively straight-forward probabilistic reasoning, the chance of a green light in  $t_p$  seconds, given a green at current time  $t$  can be found to be:

$$P[\ell(t + t_p) = g | \ell(t) = g] = \begin{cases} \frac{t_g - t_m}{t_g} & t_m \leq t_r, \quad t_m \leq t_g \\ \frac{t_g - t_r}{t_g} & t_r \leq t_m \leq t_g \\ 0 & t_g \leq t_m \leq t_r \\ \frac{t_m - t_r}{t_g} & t_g \leq t_m, \quad t_r \leq t_m \end{cases} \quad (2.5)$$

where  $t_m = \text{mod}(t_p, t_g + t_r)$  is the residue of division of  $t_p$  by  $t_g + t_r$ . In other

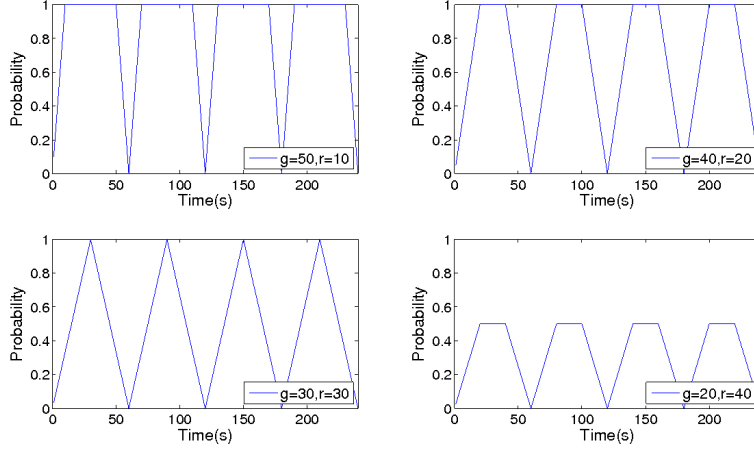


Figure 2.3: Conditional future probability of green given that the light is currently red, for four different light timing patterns. In all patterns the total cycle time is 60 seconds, with the lengths of green and red indicated in the legends. The time axis is  $t_p$  as described in Equations 2.5 and 2.6.

words, because the signal clock is assumed to be periodic, the resulting conditional probability is also going to be a periodic function of time with the same period. Similarly, the chance of a green light in  $t_p$  seconds, given a red at time  $t$  is:

$$P[\ell(t + t_p) = g | \ell(t) = r] = \begin{cases} \frac{t_m}{t_r} & t_m \leq t_r, \quad t_m \leq t_g \\ 1 & t_r \leq t_m \leq t_g \\ \frac{t_g}{t_r} & t_g \leq t_m \leq t_r \\ \frac{t_g + t_r - t_m}{t_r} & t_g \leq t_m, \quad t_r \leq t_m \end{cases} \quad (2.6)$$

Figures 2.2 and 2.3 show several probabilistic prediction examples with different splits between red and green but with the same cycle length. These are visualizations of the probabilities used in the probabilistic simulation cases described next.

## 2.5 Simulations

A series of simulations were run in increasing order of complexity and increasing applicability to real world applications. The first step involved a simple case study, whereby the efficacy of the algorithm was examined in a generic scenario. The second step was to implement the algorithms as part of a Monte-Carlo simulation, whereby in each simulation the consecutive signal timing configurations were randomly adjusted to simulate spatio-temporal effects similar to that of varied intersection geometries. In the third step, traffic signal spacing and timing configurations were adapted from a semi-urban environment, further validating the applicability of the algorithms.

We evaluate three levels of SPAT information (none, deterministic, and probabilistic) in all of the following studies. In many of the simulations, a vehicle which is unaware of the future phase of traffic signals would have to alter the vehicle velocity for some of the traffic lights and stop at some of them. In many of the simulations, a driver with full SPAT information and sufficient space and time is able to avoid coming to a stop at any of the traffic signals. The real time information case, with probabilistic models, often falls somewhere in the middle.

In the scenario with probabilistic information, the optimal control problem in Section 2.3.2 was solved in a receding horizon manner and once per sample time using DDP. To simulate the uncertainty in the phase and timing of an actual traffic signal, a random number generator could be used to slightly and randomly shift the start of a green and change phase durations. At each sample time, a prediction of the



probability of green was made for the remainder of the trip using only the current color of the lights and an assumed and fixed green/red split ratio as described in Section 2.4; this prediction was fed to the DDP algorithm at each sample time. The recent behavior history of a signal was not accounted for in the prediction stage.

A maximum speed limit of 20 meters/second is enforced in all simulations, corresponding to an arterial road. The simulated driver is required to start at zero velocity and a terminal constraint is enforced such that the driver ends at zero velocity.

In all simulations, the penalty weights in the cost function  $J$  are set equally to empirically derived values of  $w_1=1/8$ ,  $w_2=1/8$ . Weighting factors values are derived empirically such that simulated vehicles complete the distance in a reasonable time without violating red lights. This involved several, but not necessarily exhaustive, iterations. The value of  $\epsilon$  is set at  $10^{-6}$ . To solve the DDP, the solution space is discretized to distances of 20 meters, time increments of 1 second, and velocity steps of 1 meter/second. In this discretization grid choice, we have tried to maintain the computational time and memory requirements at a reasonable level without noticeably influencing the solution.

AUTONOMIE, a high fidelity vehicle simulation environment developed by Argonne National Laboratory, was used in calculating fuel economy. In simulations where it was not computationally feasible to run all cases through a full high-fidelity AUTONOMIE simulation cycle, a simplified vehicle model was developed using efficiency maps taken from AUTONOMIE and a simplified gear shifting logic <sup>4</sup>. The

---

<sup>4</sup>For example, the effects of engine start and stop transients on fuel economy were not modeled

simulated vehicle is a two-wheel-drive, automatic transmission, conventional-engine vehicle. This vehicle had a total mass of 1580 kg, an engine producing a peak of 115 kW, and a constant electrical load of 200 W. The velocity profiles generated by the dynamic program were fed to this model to calculate the fuel economy for each case. The simplified model provides a significant reduction in computational time when calculating the fuel economies for large numbers of simulation cases and we believe the fuel economy numbers will remain directionally valid, if not in the absolute sense<sup>5</sup>.

### 2.5.1 Motivating Demonstration

A case study was run as a motivating first step and involved a single simulation of a set of three consecutive signal timing configurations. The velocity profiles for the case studies with no advanced information, with read-time only information, and with full horizon information are shown in Figure 2.4. A DP solution for the full horizon information case has a smooth velocity profile, an uninformed driver must stop and start at lights, and a driver with access to real-time signal information is able to partially smooth her/his velocity profile. In Figure 2.4 the uninformed driver is required to come to a complete stop twice, for a combined total of about 7 seconds of idle time.

To help visualize how the decisions are made as the car receives more information and progresses along, time-lapses of the real-time information simulation in Figure [2.4](#) are shown in the author-derived fuel economy calculations.

---

<sup>5</sup>The directionality of fuel economy results utilizing this simplified model matches those computed using AUTONOMIE as in [19]

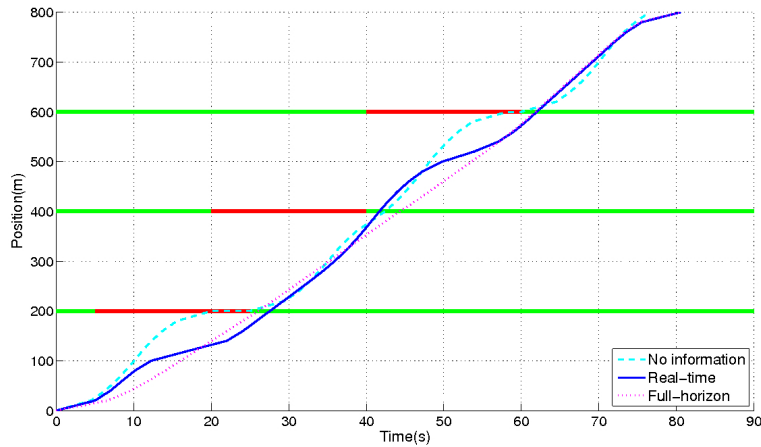


Figure 2.4: Case Study: Comparison of velocity profiles of a driver with no information, real-time information, and full horizon information.

2.4 are shown in six subplots of Figure 2.5. In these subplots the information about the future color of a light is only probabilistic and is visualized by a red to green color spectrum. As the simulated driver approaches a traffic signal, the probability prediction becomes both more confident (i.e. probability becomes bimodal around either 1 or 0), and more relevant to the simulated driver. Bright red indicates the probability of green is near 0. Bright (neon) green indicates the probability of green is near 1. Dark reds and dark greens indicate probabilities in the middle. Because probability is only one of the terms in the cost function, at times it may appear as though the driver is moving aggressively towards a light with a high probability of red; the color spectrum shown is only the value of probability of green, and does not reflect the total cost function. A simulation movie for the first case study can be found at [2].

The fuel economy for each scenario was evaluated in AUTONOMIE and using

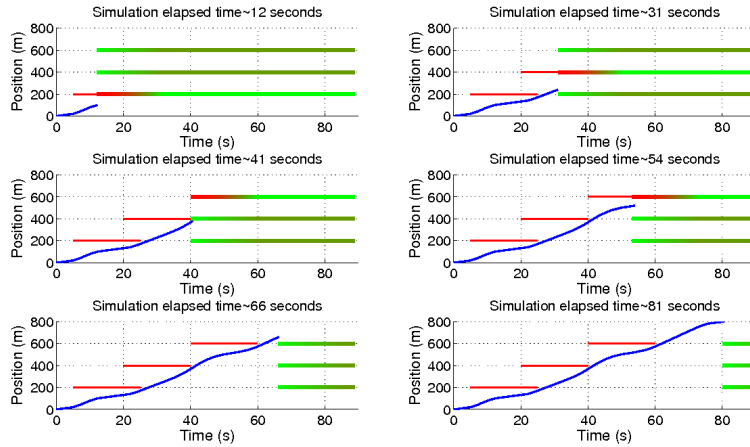


Figure 2.5: Snapshots of a running simulation for Case Study 1. Probability prediction is indicated by the red to green color spectrum. See the video at the following link [2].

Table 2.1: Fuel economies in miles per gallon (mpg). The results show the impact information can have on the energy efficiency.

	Case Study
No Information	10.3
Real Time Information	16.6
Full Information	31.5

the full vehicle model. The results presented in Table 2.1 are promising. With only real-time phase data and the probabilistic prediction model, a 61% increase in fuel economy over an uninformed driver is observed. This corresponds to 29% of the potential benefit of having full and exact future knowledge of SPAT information. Note that because the total simulation distance is only 800 meters with three signals, the fuel economy differences may be more exaggerated than average gains expected over driving cycles where traffic signals are less frequent.

## 2.5.2 Monte-Carlo Simulations

While the results of the preceding case study is promising, it is not clear if significant improvement in average fuel economy can be gained with the proposed algorithm, if relative offsets in the three signal timings are varied. In other words, it remains to verify that fuel economy gains were not solely a result of author-designed signal offsets. Therefore in this section we evaluate a statistically significant number of cases with randomly generated timing offsets; this is a variant on a Monte-Carlo experiment.

For these Monte-Carlo simulations, drivers with access to the three levels of information were run, in which the start of red phases were randomized within a window of sufficient length for the driver to complete the route. The total cycle length, and length of each red were kept constant. Also the proportion of red to green times across all simulations were constrained to be the same (this ratio is the average used for the simulated signals, and could match a 24 hour or any other temporal average for a specific traffic signal). The start of the red phase of

Table 2.2: Monte-Carlo simulation results reflect the positive influence of information, on average, on fuel economy.

	Mean(MPG)	Standard Deviation(MPG)
No Information	25.9	5.0
Real Time Information	29.9	3.7
Full Information	32.5	3.0

each signal was uniformly varied within the cycle so long as the full length of red was preserved. The start of each red of a traffic signal was chosen independently of the start of red of the next traffic signal. Three thousand simulated cases, with three traffic signals per simulation, with a simulation length of eight hundred meters were run (1000 simulations for each level of information: no information, real-time information, and full information).

The fuel economy for each of the The Monte-Carlo simulations was obtained by feeding the resulting velocity profile to our simplified vehicle model. The averaged results for each information level are summarized in Table 2.2. The results indicate that for the road conditions described and with only real-time information and the probabilistic models, an average of 16% increase in fuel economy could be expected, representing approximately 62% of the benefit of full and exact traffic signal timing information.

It was determined that the chosen 1000 simulations per information level is sufficient and captures reasonably well the average fuel economy for each information level. This can be seen in Figure 2.6 where the cumulative average fuel economy over the number of simulations is shown for each information level. By the time

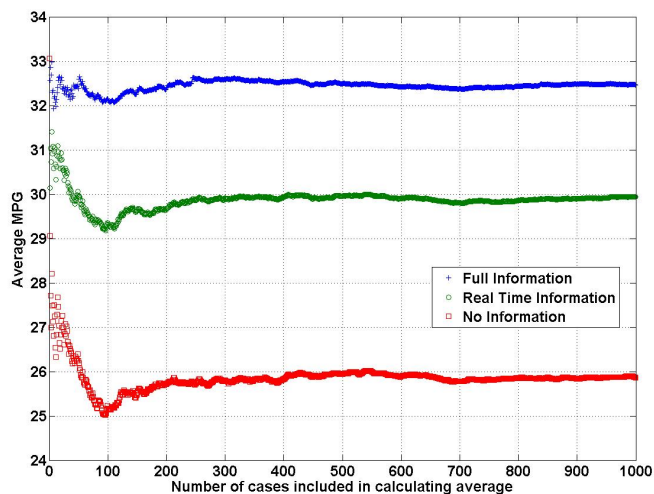


Figure 2.6: Average fuel economy for increasing number of simulated streets.

1000 runs were simulated for each level of information, only very minor changes in average MPG occurred with the addition of more cases.

Figure 2.7 shows histograms of Monte-Carlo fuel economy traces. As apparent in the figure and also shown in Table 2.2, the standard deviation of the case where the driver has no information is highest, where the case where the driver has full information is lowest. One possible explanation of this is that the driver with no information has significantly different fuel economy when the driver has to stop at all lights versus no lights, whereas the driver with full information is able to achieve more uniform fuel economy results by smoothing the velocity and avoiding stopping.

The driver with no information is occasionally able to be faster and more efficient than the driver with some or full information. This situation may occur as a result of the no-information driver accelerating at full speed, and just passing before a

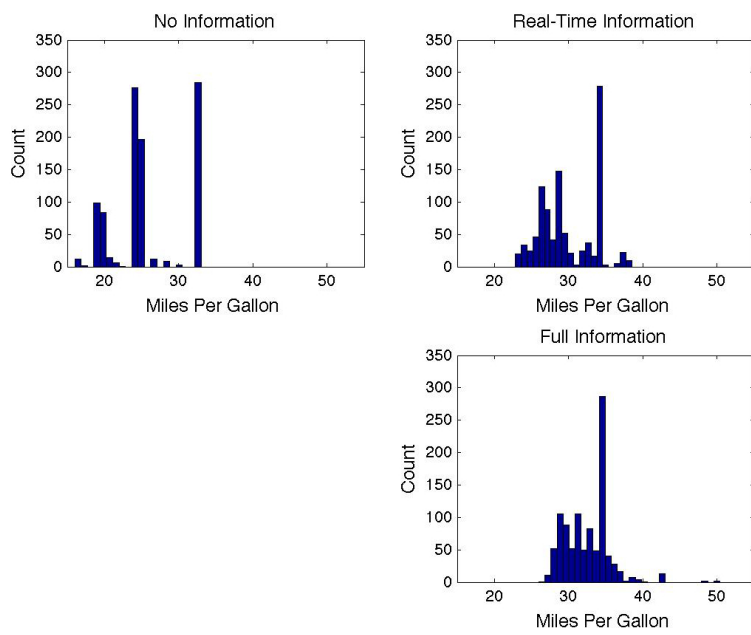


Figure 2.7: Histogram of Monte-Carlo simulation results for the three levels of information.



light turns red. When this occurs, the driver is able to significantly reduce total trip time, which in turn reduces fuel usage. Because the real time or full information drivers have cost functions (Equations 2.1 and 2.4) which place some weight on acceleration, the optimal solution is rarely maximum acceleration. The driver with no information applied full acceleration and full deceleration as necessary, reflecting an aggressive driver.

### **2.5.3 Simulations Using Recorded Timings from Arterial Adaptive Lights**

With Monte-Carlo simulations indicating positive relationships between future information about traffic signals and fuel economy, we turn to more realistic examples in which we use signal timing and geometry of actual intersections. Toward this goal, we have obtained the timing of signals of three intersections from the city of Fremont in California. The lengths of the green phases in the direction of travel for these simulations for a 24 hour period can be found in Figure 2.8. From this Figure, it is clear that the traffic signals are not fixed, they respond to traffic conditions. Use of actual signal timings and actual offsets between multiple intersections, reduces any unintended bias that may have been present in our Monte Carlo simulation design. Moreover, we show in this section that while our proposed algorithms were developed for fixed-time signals, they are robust to variance in nominal traffic signal timing and could potentially be used even in the presence of actuated traffic signals.

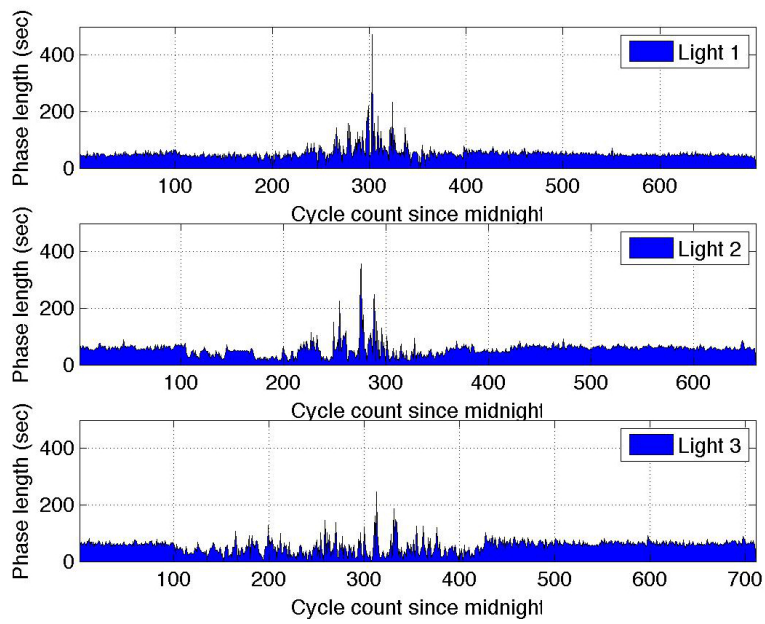


Figure 2.8: Histories of relevant phases for each light along the chosen real-world route, for every cycle over a 24 hour period (midnight to midnight).

A vehicle was simulated driving through the three traffic signals every 10 minutes over the 24 hours yielding a total of 144 simulated drives per level of information. The real-world distance between the signals is preserved in the simulation, such that the simulated vehicle has to cover the same distance using the same traffic signal timing offsets as a real driver would encounter. The total simulation distance is 1320 meters. The first light occurs 520 meters into the simulation. The second light occurs 280 meters later. The third and final light occurs at 1200 meters from the start. The resolution of the dynamic programming algorithm was kept similar to simulations in Section 2.5.2. Velocity resolution remains at 1 meter per second, distance resolution remains at 20 meters, time resolution remains at 1 second. The traffic signal phase and timings are taken from recorded data from the city and are merely played back into this simulation. No other vehicles are considered to be on the road - the only obstacles the vehicle routing algorithm must avoid are the lights themselves. For the purposes of prediction, the real-time simulation is given a 24 hour average of red and green lengths, though if more relevant averages (for example a short-term average, a time of day average, or other statistical means) are available, they may continue to improve the performance of this real-time information case.

Fuel economy was calculated in the same manner as the Monte-Carlo simulations, due to the calculation time of AUTONOMIE for the number of drive cycles considered. The results of these fuel economy calculations can be found in Figure 2.9, and the means of those simulations found in Table 2.3. The maximum error between the velocity profile generated and the velocity profile followed by the fuel economy calculations was 3.6%. Figure 2.9 and Table 2.3 both confirm the positive

Table 2.3: Fuel economy results from recorded real-world traffic signal timings with simulated vehicles moving between the lights reflect the positive influence of information.

	Mean(MPG)	Standard Deviation(MPG)
No Information	31.7	3.1
Real Time Information	33.7	3.0
Full Information	34.5	3.6

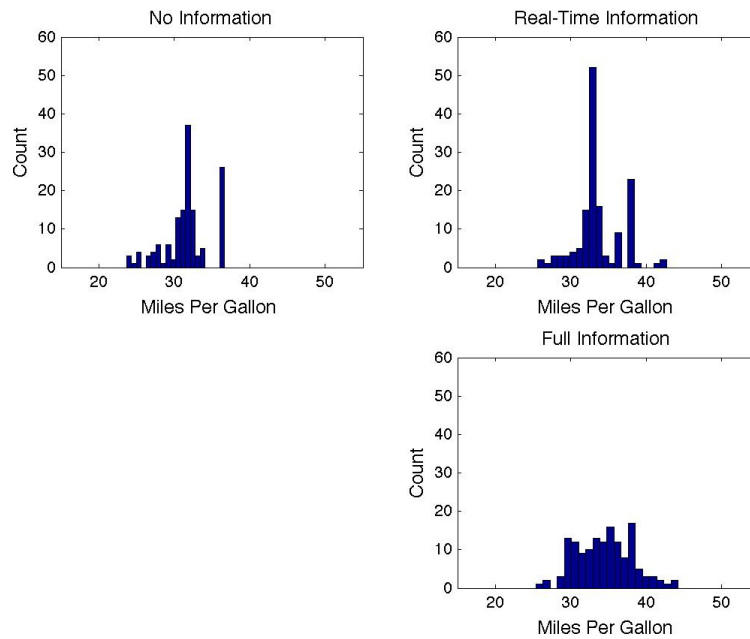


Figure 2.9: Histogram of fuel economies for real-world traffic signal timings.

influence of information on fuel economy.

The real-world data used in Section 2.5.3 provides another step towards implementability in comparison to the Monte Carlo simulations of Subsection 2.5.2. The use of recorded traffic signal data reduces the possibility of author-induced errors, e.g. inadvertently creating red or green waves. The simulation indicates that drivers with access to real-time information were able to improve fuel economy over drivers with no information by approximately 6%. This accounts for roughly 70% of the potential gains available through access to full and exact future knowledge of traffic signal timing. The particular implementation chosen here allows finding the new optimal trajectory based on the most current data by recalculating the cost to go and control matrices at each position step. This is one technique for dealing with unexpected traffic, pedestrians crossing out of cross-walks, and other disturbances.

In the real-world data, at some times throughout the day the base timing plan of some of the traffic signals have cycle lengths of 60 seconds. The decision to simulate a vehicle every 10 minutes (a multiple of the 1 minute nominal cycle length) is therefore a potentially problematic choice. However, as the traffic signals are adapting to traffic conditions, the cycle length and splits adjust, reducing the chance of aliasing of results. This ensures that the simulations are not 144 repetitions of the same cycle (the variation in simulations can be confirmed by reviewing Figure 2.9). Additionally, the timing plans of the various traffic signals appear to be lacking in synchronization, as even late at night and early in the morning, when neither vehicles nor pedestrians make calls, the lights drift with respect to each other.

In general, in a coordinated series of signals, under actuated or adaptive control

logic, the offsets and other signal timing parameters may play an increasingly important role - for example in creating a green wave. If the proposed probabilistic models are able to assist a driver in joining a green wave, this may have a positive effect on fuel economy for that driver.

## 2.6 Conclusions

This chapter statistically evaluated velocity planning algorithms which minimize idling time behind red lights based on probabilistic traffic signal timing models that we proposed. Three cases were evaluated - vehicles with no information about upcoming traffic signals, vehicles with real-time information, and vehicles with full and exact future knowledge of traffic signal timings. Drivers today fit into the first case - the least efficient. Drivers of the future may fit into the third and most efficient case, if infrastructure and technologies develop to provide this information. The middle case is feasible today, and obtains much of the potential benefit obtainable via knowledge of upcoming traffic signal timing.

Real time knowledge with probabilistic models where the driver encountered fixed-time lights yielded an optimistic 61% increase in a motivating case study, and a 16% increase in average fuel economy across 1000 multi-signal simulations of fixed time signals. The same models in combination with real time information yielded a 6% increase in fuel economy for actuated signals. These reflect technologies which could be feasibly implemented with little or no infrastructure changes and with only software updates to current production vehicles.

## 2.7 Acknowledgement

The authors thank Dr. Chen Zhang for his assistance regarding shaping the DP discretization. This work was supported by the National Science Foundation grant number CMMI-0928533, and by BMW Group.

## Chapter 3

# Machine Learning Prediction of Actuated Traffic Signal Phase and Timing Data

### 3.1 Abstract

Predictive Traffic Signal Phase and Timing (SPaT) information has the potential to improve in-vehicle driver assistance and safety applications, increasing safety and energy efficiency. However, due to the diverse age and brands of technologies implemented in traffic signal controllers, and the non-deterministic influence of vehicle and pedestrian arrival, prediction of the next active phase(s) and phase length is a challenging problem. This chapter evaluates Transition Probability Modeling, Decision Tree, Multi-Linear Regression, and Neural Network machine learning methods



for use in the prediction of traffic signal phase length. Final models used the attributes elapsed phase length, a three cycle mean, a ten cycle mean, the time vehicles have been waiting at inactive phases for all 8 phases, and the vehicle call status for all 8 phases, in both training and testing the algorithms. Multi-Linear Regression performs best in the tests performed herein, with a root mean square prediction error, averaged across 8 phases over 24 hours of test data, of 3.6 seconds.

## 3.2 Introduction

Fully autonomous vehicles have the potential to significantly improve vehicle safety and energy efficiency, but both public acceptance and technological solutions are currently insufficient. Consumer reception of driver assistants like navigation systems is popular and increasing, and may be one way to increase vehicular safety without giving up driver volition. However, while academic researchers and consumer products have numerous offerings which all guide towards avoiding traffic congestion, none of these programs utilize traffic signal information and as a result many of them have mediocre urban travel time estimations. The missing link is the availability of traffic Signal Phase and Timing (SPaT) information.

Prototype vehicles and infrastructure built by the Vehicle Safety Communications Consortium (Ford, GM, Honda, Mercedes-Benz, Toyota) have shown it to be possible to reduce collisions at intersections through red light runner prediction [26], as have others [27]. In many sources [11, 19, 28, 29, 30] it is shown to be possible to improve fuel economy through appropriate speed recommendations. These ap-

plications generally assume the presence of real time and or future traffic signal information; this is a large assumption because knowledge of current and future SPaT information is an undeveloped area.

Microsoft [31] has worked on traffic flow prediction, using mobile phone locational data, to observe and predict current and future traffic flow information. Similarly, University of California Berkeley PATH program [32] have been able to use cell phones to build traffic flow models for travel time prediction. Researchers from MIT, and the University of Luxembourg have created traffic congestion models and routing algorithms which avoid these hot spots in [33, 34]. Finally, the company WAZE is built primarily around a smart-phone application which builds predictive congestion models [35]. All of these focused on traffic flow, not signal SPaT calculation or prediction.

In [13], current traffic signal status is determined via mobile phone cameras placed in vehicles. From there, the phones transmit current status of signals, and make predictions about future states of the system. However, the signals in [13] had only 2 phases, and therefore the results reflect an unrealistically simple and optimistic situation if the algorithms are intended to be utilized generally.

In [12], traffic signal data is collected from the city, both static location and parameter data, as well as dynamic actuation and timing data. This solution, when this data is available, has many benefits - high accuracy, low latency, and immediate knowledge of outlier events like emergency vehicles and sports games. An additional benefit is that clock drift in pre-timed signals is negated by the constant connection to the data source. However, accurate predictions are only available for pre-timed

signals (not for actuated or adaptive), and transitional periods still present an issue for prediction.

In [18, 36], multiple vehicle locational and velocity data traces are used for back-calculation of pre-timed signal timings in two complementary techniques. However, neither successfully addresses actuated or traffic adaptive signals.

The most relevant source is a master's thesis from the Technical University of Munich which uses machine learning methods to predict both phase lengths and the next phase [4]. However, because the metric for success also includes time periods when the signal is inactive, the metric is overly optimistic and hides the accuracy of the system in predicting active phase lengths. The use of this success metric unfortunately does not allow direct comparison therefore between the results presented and those found in this manuscript. Additionally, the thesis does not provide comparison across methods for the same prediction - the different methods are used for different types of prediction and therefore no objective analysis of the relative applicability of any method is available.

While these solutions may provide access to real-time SPaT information, they are not able to provide *future* traffic signal information. For applications like route planning, engine start-stop, and transmission control, future knowledge of traffic signal information could improve the overall vehicle efficiency.

The goal of this chapter is to provide a comprehensive analysis of several methods performance at predicting the length of a traffic signal phases. In addition, it is important to provide a measure of confidence in that prediction to determine its applicability in a particular driver assistance system.

This paper is organized as follows: in Section 3.3, a brief overview of the basic operation and types of traffic signals is presented. The process of obtaining historical data, the attributes used in prediction, is described in Section 3.4. Preprocessing and a preliminary analysis to determine suitability to data mining techniques can be found in Section 3.5. In Section 3.6, probabilistic models are used to predict pre-timed signals. For the actuated and adaptive signals in Section 3.7, data mining methods, and motivation for their use in prediction of SPaT information are presented and evaluated. Discussion of results, comparison across phase length prediction methods, and future work are presented in Section 3.8.

### 3.3 Background on Traffic Signal Control

An overview of a standard intersection may be found in Figure 3.1.

Some terminology is an important starting point for understanding traffic signal controller logic, though the terminology varies from manufacturer to manufacturer and city to city. The word *phase* is commonly used to describe a permitted movement, be it vehicles in the through direction or a turning direction, or pedestrian movements. Phases are *active* during the time period that specific movement is shown a green light or walk symbol (sometimes active phases includes yellow). It is common to assign even phase numbers to through lanes, and odd phase numbers to dedicated left turn lanes. Permissive right turns are generally not given their own phase number, it is considered to be lumped with the through phase. Similarly, pedestrian movements are generally numbered in accordance with the appropriate

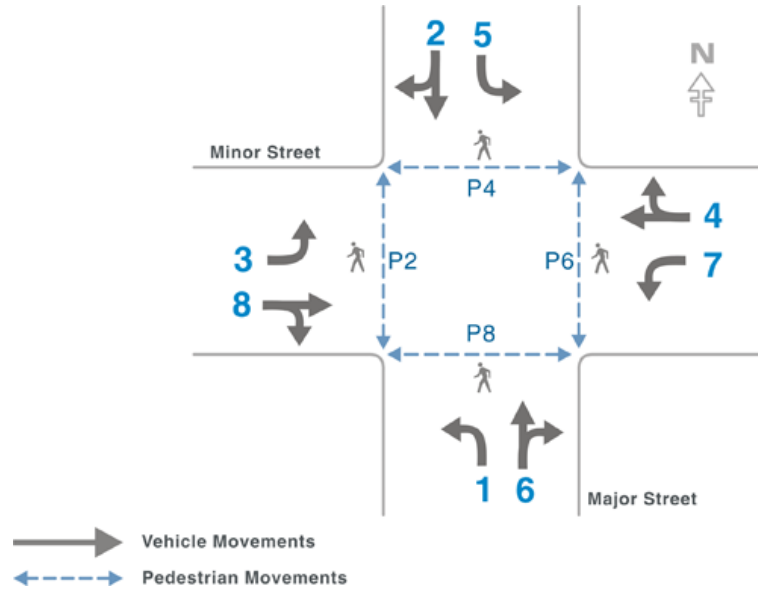


Figure 3.1: Phase numbering of a standard intersection per FHWA standards [3].

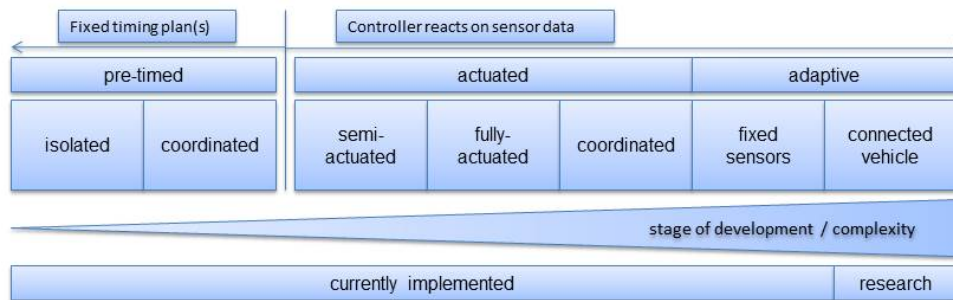


Figure 3.2: Overview of signal control schemes in place around the United States.

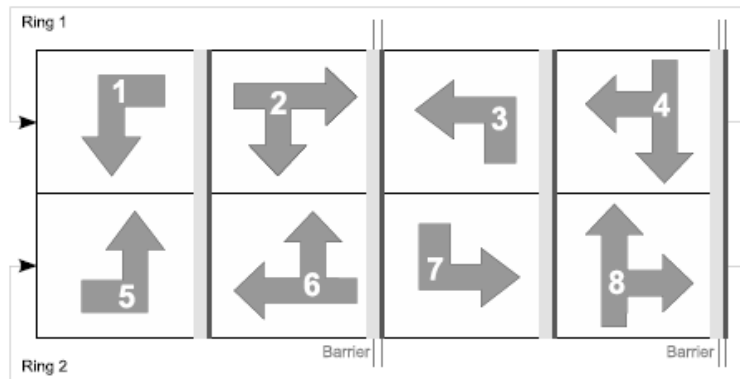


Figure 3.3: A standard (leading left) ring-barrier diagram [4].

phase which would occur concurrently (e.g. westbound through lane for vehicles, and westbound pedestrians are normally numbered the same, as seen in Figure 3.1).

Phases are commonly grouped into organizational structures called *rings*. The phases of an 8 phase intersection will often be grouped into 2 rings, such that phases 1-4 and 5-8 are grouped into rings 1 and 2 respectively, as seen in Figure 3.3. In order to ensure the safe passage of vehicles while concurrent movements are active, *barriers* are used to separate conflicting movements (sometimes called all-red periods). The ring-diagram for an 8-phase intersection may be found in Figure 3.3. In this example, both phases 1 and 5 could be activated at the beginning of the cycle. If either active phase *gaps out* or reaches the *maximum green*, through a lack of vehicle movement or reaching the longest programmed active phase duration allowed, that phase will transition to the next phase in that ring, phases 2 and 6 respectively. It is therefore possible for the combinations 1 and 5, 1 and 6, 2 and 5 or 2 and 6 to be active concurrently, but no other combinations. Once **both** 2 and

6 have either gapped out or maxxed out, they become inactive, the barrier may be safely crossed, and phases 3 and 7 are activated. The rotation through the phases of the intersection, permitting movement such that all requested phases are served is also sometimes called a *cycle*.

When a vehicle arrives at an intersection, there is often a sensor (commonly an inductive loop), which tells the signal controller that there is a *vehicle call* requesting activation of that phase. This is essentially equivalent to a pedestrian pushing the crosswalk button, which places a *pedestrian call* to the controller, requesting activation of that phase.

Numerous control strategies have been developed to efficiently process traffic at varying intersection geometries under changing conditions, as can be seen in Figure 3.2. For the purpose of prediction, the primary difference is whether the intersection controller operates in a fixed (cyclical) pattern, or whether the signal responds to vehicle requests. This distinction discriminates pre-timed signals from actuated and adaptive signals. Therefore, we describe the simpler prediction of pre-timed signals in Section 3.6 and the more complex actuated and adaptive signals in Section 3.7.

In this chapter, it is assumed that intersections are of the same geometry as found in Figure 3.1, with a nominal count of 8 vehicular phases. For intersections which use fewer phases, the same model may be used with some columns empty.

## 3.4 Raw Data Acquisition

The process used to acquire the data used in building the predictions models in Section 3.7 are described here.

First a note on terminology; we use the word *tuple* to describe a set of attributes which are connected (generally a single record). These attributes, described in Section 3.4.1, form the basis of the historical information stored in a database from which predictive models are built.

Data in this chapter was recorded in Portland, Oregon, San Jose, California and Fremont, California. Traffic signals in all three test areas have been connected to controllers, in this case Siemens controllers. Those controllers are in turn networked to Traffic Management Centers (TMC). Each TMC is able to poll its respective traffic signals about their current status. The software program WireShark was used to replicate the data coming in to the TMC, and the duplicated data stream was sent to and recorded by a separate server for storage and analysis [37]. In each city, data was collected over two consecutive business days, with the first day being used for training data and the second day being used for test data.

The training data in Fremont provided 100351 training tuples. The test data contains 97703 tuples for classification. The training data in Portland provided 15922 training tuples. The test data contains 15626 tuples for classification. The training data in San Jose provided 31012 training tuples. The test data contains 32195 tuples for classification. The apparent difference in data sizes is due to recording format, as Fremont training data was recorded at 1Hz independent of status updates (a limita-



tion imposed by the controllers), whereas Portland and San Jose are change-based recording. A visualization of the phase lengths, taken from the Fremont training data, may be found in Figure 3.4.

Descriptions of the recorded data, later used as attributes, can be found in Subsection 3.4.1.

### 3.4.1 Raw Data (Attributes)

While the raw data recorded directly from the traffic signal is useful, modification of that data as well as computation of new attributes has the potential to significantly increase the accuracy of the data mining models and techniques. What we describe here are the various data sets from which we pulled our model training and testing data. Not all of these attributes will prove useful, not all of these attributes will be included in the final model, and not all of these attributes were present at all signal controllers in all testing locations.

- *phaseOnStatus* is the encoded phase status of all phases of a signal.
- *ring1Status* is the phase status for all phases of ring 1 of a signal.
- *ring2Status* is the phase status for all phases of ring 2 of a signal.
- *ring1Walk* is the walk status for all phases of ring 1 of a signal.
- *ring2Walk* is the walk status for all phases of ring 2 of a signal.
- *vehicleCall* is the vehicle call status for all phases of a signal.

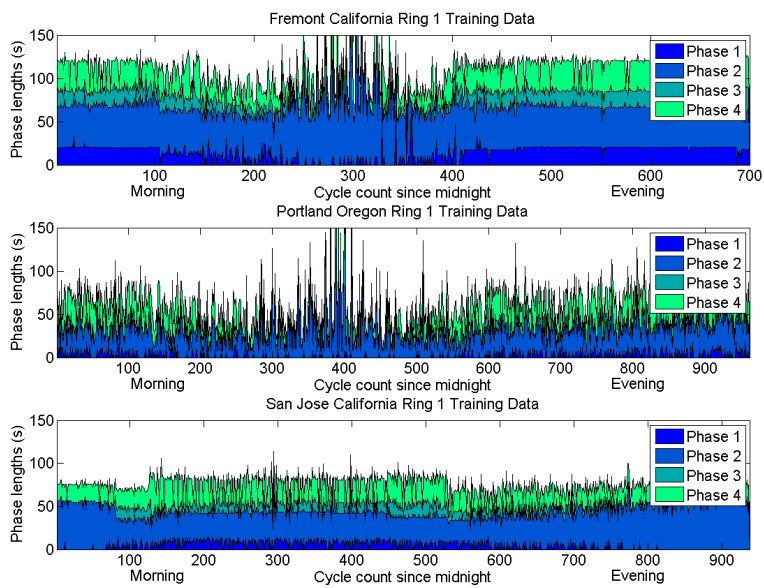


Figure 3.4: Twenty four hours of training data for an example signal at each of the three test locations. Graphs show only data from Ring 1, and have been abbreviated in the Y-axis, cutting off outliers in order to show the phase length variance in the majority of cycles.

- *pedCall* is the pedestrian call status for all phases of a signal.
- *coordinationInControl* is a flag indicating whether the master intersection is attempting to coordinate the corridor.
- *elapsedTime* is the time, in seconds, since the previous midnight.
- *elapsedPhaseLength* is the time, in seconds, since the last phase change.
- *waitTime* is the time, in seconds, since the vehicle loop detector indicated a vehicle arrived at an inactive phase (red).
- *shortMean* is the mean time, in seconds, the phase lasted over the last 3 cycles.
- *longMean* is the mean time, in seconds, the phase lasted over the last 10 cycles.
- *remainingTime* is the time, in seconds, until the phase ends.

Many attributes were modified for programmatic reasons, but no information was lost or gained. Additionally, the attributes *elapsedTime*, *elapsedPhaseLength*, *waitTime*, *shortMean*, *longMean*, and *remainingTime*, were calculated and appended to each tuple entry, and are not generated by the traffic signal controller software. A full 24 hours of data per signal is approximately 5 megabytes; for a city the size of San Jose, approximately 3 gigabytes of data is recorded per day.

Of special note is the attribute *remainingTime*; this is calculated on historic data. This will be used as a “ground truth” for the supervised learning aspect of the data mining.

## 3.5 Data Preparation and Exploration

It is important to determine which variables are significant predictors of the phase length, as computational requirements will increase commensurate with an increase in modeled data. Visual inspection indicates that the individual phase lengths are fairly consistent in the early morning and the evening. During mid day, the traffic signal controller switches from coordinated action with the other signals on the street to operating individually; the variance of individual phase lengths increases significantly when this occurs.

We use here Principal Component Analysis to determine which attributes are responsible for the most variance in the system, and therefore which attributes to build into predictive models first. An example attribute has been selected for additional variance analysis to show the applicability of that attribute in statistical predictive models.

### 3.5.1 Principal Component Analysis

Principal Component Analysis is a transformation of a set of attributes into a linearly uncoordinated set. This leads to an equal or lesser number of vectors, called the Principal Components. It is, for the purposes here, a multi-dimensional rotation of axes such that variance is maximized along axes. Each column (attribute) had the column-mean removed, a process called “centering” which standardizes attribute ranges. The axes are then ranked according to total variance along that axis. Axes which have relatively low variance are candidates for removal (model simplification).

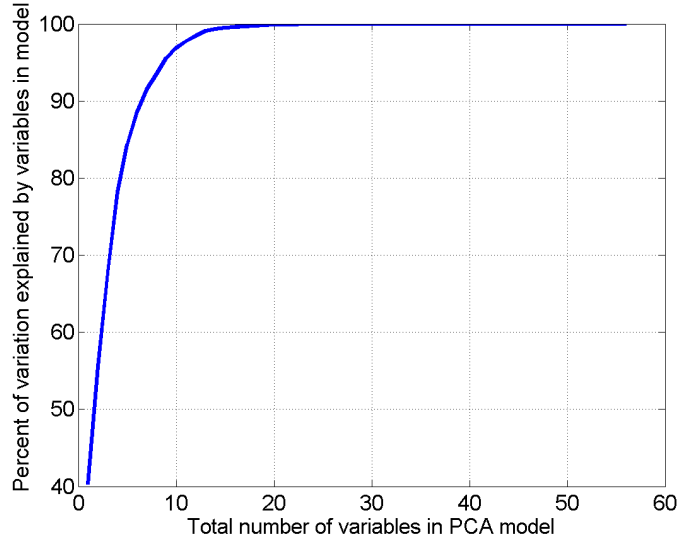


Figure 3.5: PCA results.

The PCA results found in Figure 3.5 indicate that the most influential variable accounts for only 23% of the variance of the system. It also implies that many variables have extremely low variance in the system and could be dropped with little statistical significance in the results. This information was used to rank which variables should be initially included, in order to build computationally efficient models. The final list of attributes included in the models may be found in Section 3.8. We turn now to a detailed analysis of one attribute to determine its potential applicability to prediction models.

Table 3.1: Standard Deviation (in seconds) of Training Data

	Phase 1	Phase 2	Phase 3	Phase 4
$\sigma$ for 24 hr period	4.85	24.14	5.15	9.78
$\sigma$ for morning, CIC ON	2.09	4.78	2.75	5.05
$\sigma$ for midday, CIC OFF	3.12	35.24	5.88	10.97
$\sigma$ for evening, CIC ON	2.44	5.27	3.57	4.56

### 3.5.2 Variance of Phases

The attribute *coordinationInControl* (CIC) was selected for additional observation, as this attribute explained a significant amount of variability. This attribute is binary valued, representing whether multiple signals are being coordinated. Each signal controller may have several algorithms which determine phase lengths throughout the day. These algorithms may be specific to on- and off-peak rush periods. At certain times of the day an individual signal controller may have the ability to operate freely, and at other times of the day a master controller may be attempting to guide a platoon of vehicles through a series of coordinated signals.

The data set was divided based on that variable, and the standard deviation was observed. The results can be found in Table 3.1, showing the promise of a statistical method; by differentiating according to whether the signal is able to operate freely or must abide by coordination, the standard deviation in most cases drops significantly.

## 3.6 Prediction of Pre-Timed Signals

As indicated in Section 3.3, in this chapter we divide the various types of traffic signal controller logics into two areas - those that react to traffic, and those which do not. This section is on pre-timed signals, which do not react to current traffic conditions, and ostensibly follow a predetermined pattern of phase lengths. However, there are some complications. There can be much uncertainty in the phase and timing of a traffic signal which makes predicting its future state quite challenging. For fixed-time traffic signals which do not respond to traffic conditions and operate only on a timing table, we have confirmed the finding that the traffic signal clock drifts some during a 24 hour period (see [38, 13] for more details on variation from the pre-timed schedule). There is also uncertainty during the time periods around timing schedule shifts, when the controller is transitioning from one schedule to another.

Due to the aforementioned uncertainties, even for fixed-time signals it is not possible to determine the start and duration of greens deterministically. Therefore in this section we propose a probabilistic prediction framework to handle the case with partial or uncertain information. We focus on cases where only i) the current phase (color) and ii) the average red and green lengths for a signal are known. We use this information to predict the probability of a green over the planning horizon.

As previously mentioned, access to the current phase of the traffic signal is a major technological hurdle. However, solutions have been proposed and implemented in [12, 22, 39] that could address this problem. Other approaches, including those that rely on Dynamic Short Range Communication (DSRC), can be found in [11, 23, 24].

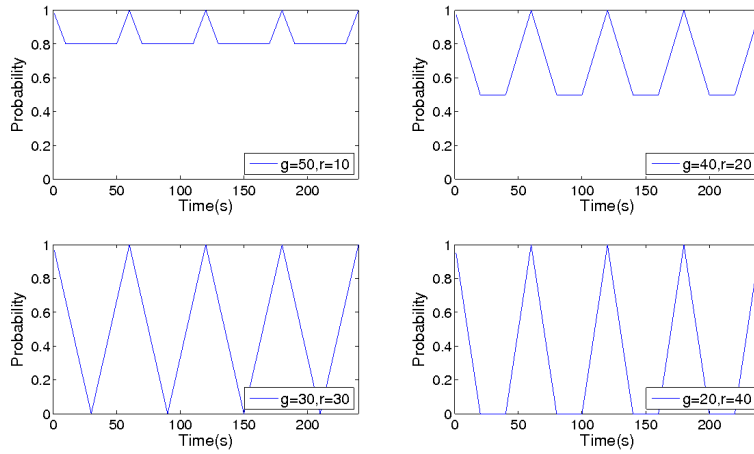


Figure 3.6: Conditional probability of green given green now, for four different light timing patterns. In all patterns the total cycle time is 60 seconds, with the lengths of green and red indicated in the legends. The time axis is  $t_p$  as in Equations 3.1 and 3.2.

Let us denote the state of a light by  $\ell(t)$  which can assume two values,  $g$  and  $r$ , representing green and red respectively. We are interested in determining the probability of a light being green at time  $t + t_p$  conditioned on its current color at time  $t$ . To form this conditional probability function, we assume the durations of green and red are known to be  $t_g$  and  $t_r$  on average. We also assume the traffic signal operates cyclically<sup>1</sup> and as a result the total cycle time is fixed and equal to  $t_g + t_r$ . Using relatively straight-forward probabilistic reasoning, the chance of a green light in  $t_p$  seconds, given a green at current time  $t$  can be found to be:

---

<sup>1</sup>This is true for many traffic signals; even many of those that react to traffic have a fixed cycle, although in some cases the length may be shortened or extended.



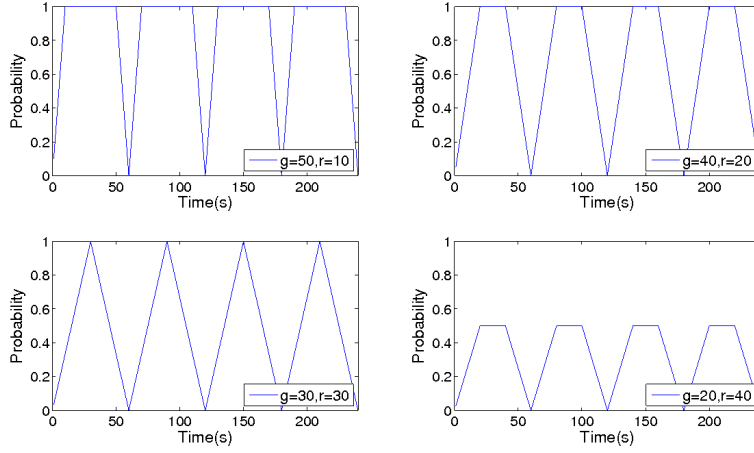


Figure 3.7: Conditional probability of green given red now, for four different light timing patterns. In all patterns the total cycle time is 60 seconds, with the lengths of green and red indicated in the legends. The time axis is  $t_p$  as in Equations 3.1 and 3.2.

$$P[\ell(t + t_p) = g | \ell(t) = g] = \begin{cases} \frac{t_g - t_m}{t_g} & t_m \leq t_r, \quad t_m \leq t_g \\ \frac{t_g - t_r}{t_g} & t_r \leq t_m \leq t_g \\ 0 & t_g \leq t_m \leq t_r \\ \frac{t_m - t_r}{t_g} & t_g \leq t_m, \quad t_r \leq t_m \end{cases} \quad (3.1)$$

where  $t_m = \text{mod}(t_p, t_g + t_r)$  is the residue of division of  $t_p$  by  $t_g + t_r$ . In other words because the signal clock is assumed to be periodic, the resulting conditional probability is also going to be a periodic function of time with the same period. Similarly, the chance of a green light in  $t_p$  seconds, given a red at time  $t$  is:

$$P[\ell(t + t_p) = g | \ell(t) = r] = \begin{cases} \frac{t_m}{t_r} & t_m \leq t_r, \quad t_m \leq t_g \\ 1 & t_r \leq t_m \leq t_g \\ \frac{t_g}{t_r} & t_g \leq t_m \leq t_r \\ \frac{t_g + t_r - t_m}{t_r} & t_g \leq t_m, \quad t_r \leq t_m \end{cases} \quad (3.2)$$

Figures 3.6 and 3.7 show several probabilistic prediction examples with different splits between red and green but with the same cycle length. While not directly comparable to a single numeric prediction as found in the following section, these models have been used in speed recommendation algorithms with success [28], and may be useful in other driver assistance applications.

The preceding probabilistic models have two downsides - they work best for fixed-time traffic signals, and because the output is a probability map and not an integer it is difficult to use as an input to other in-vehicle systems. We shift now to providing a single prediction of remaining phase length, and tackle the additional complexity of predicting the phase and timing of actuated traffic signals, in the next section.

### 3.7 Prediction of Actuated and Adaptive Signals

In comparison to the pre-timed signals in Section 3.6, in this Section we predict phase lengths for traffic signal controllers which respond to current traffic conditions.

Simple preprocessing analysis of small subsets of the data indicated that a data mining approach was promising (see Section 3.5 Subsection 3.5.2). Moving beyond

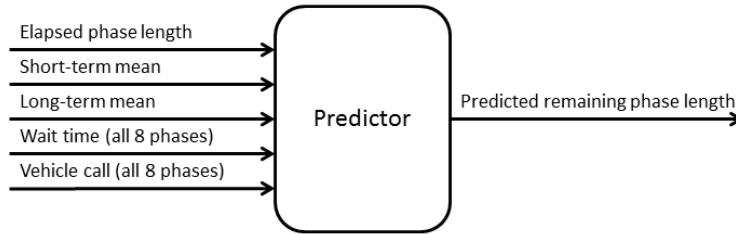


Figure 3.8: Inputs and outputs of machine learning prediction models.

simple variance analysis, more advanced data mining methods offer numerous tools for analysing and predicting future trends in time series. Four methods have been selected as most promising: Transition Probability Models (TPM), Decision Trees (DT), Multi-Linear Regression (MLR) and Neural Networks (NN).

As shown in Figure 3.8, the final set of inputs used in each method were *elapsedPhaseLength*, *shortMean*, *longMean*, *waitTime* (for all 8 phases), and *vehicleCall* (for all 8 phases), in both training and testing the algorithms. This is a total of 19 inputs, for each output. The output was an integer, the predicted remaining phase length. When a full 24 hours of data is used, the inputs and outputs are fed into the models as vectors.

In each of the following cases, the analysis was run for each phase independently, to determine if the predicted remaining phase length matched the *remainingTime* calculated from historical data. The *remainingTime* is used as the “ground truth”.

### 3.7.1 Transition Probability Modeling Approach

A phase transition probability curve is a historical record of the probability of a phase ending after a certain phase length. It is a  $1 \times p$  vector, where each entry represents the probability of the phase transitioning at  $p$  seconds. It is created by iterating through all historical phase records (tuples) (with a total of  $n$  records), and for each record, decrementing the  $p$ th vector entry and all higher entries, where  $p$  is the recorded phase length, by  $1/n$ . This represents the (historical) probability of the signal remaining active over a time window. This is a step away from the probabilistic methods in Section 3.6, towards building a single numerical predictor as might be necessary for an engine start-stop algorithm. This offers an interesting and unique method, which is very fast computationally, by comparison to the following methods.

For example, in the transition probability map in Figure 3.9 it is clear that the phase always remains active longer than 7 seconds. This is a useful piece of information. However, this map offers much more, as the largest (absolute value of the) derivative of this transition probability map is the time of highest likelihood of phase transition. It is important to note that this is a changing window - if the cycle time for the largest derivative has already been passed, the largest derivative in the remainder of the historically observed window is used. In this model we discretize into 1 second increments, then build transition probability maps on-line, adjust the window to take into account the elapsed phase length, take the derivative of the remainder of the window of the map, and find the absolute value of the maximum. This is the predicted remaining phase length.

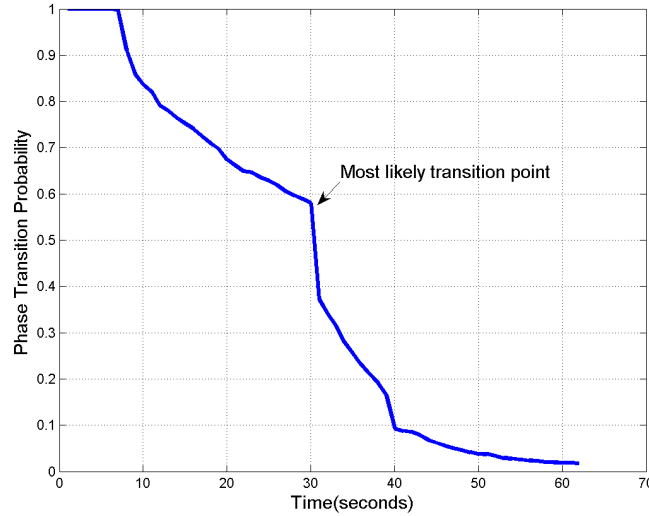


Figure 3.9: Transition probability for Phase 4 of Signal 0001 in Portland, Oregon based on data collected 09/24/2012.

### 3.7.2 Decision Tree

An implementation of decision trees was used as a starting point because the original coding structure of traffic signal controller logic is a series of logical statements. The structure of a decision tree algorithm may be able to recreate those logical statements from the controller without having access to the original controller code. We used a standard greedy decision tree, with pruning enabled, and regression at each leaf. The Gini-Simpson Diversity Index is used to determine whether to split. Error-based pruning is done after the tree is generated, and is the process of going through each node and replacing it with the class of highest probability; if this does not significantly affect the error rate, the change is kept. This both simplifies the classification tree and reduces the likelihood of overfitting to an outlier.

In the decision tree models, both the training and testing data was discretized into 1 second increments, and all phase lengths observed in the training data were used as potential classes. Test data was then classified. The difference between the classes was the error (in seconds).

### 3.7.3 Multi-Linear Regression

Multi-Linear Regression, similar to simple regression, is an attempt to create a linear equation which fits a set of 2 or more explanatory attributes or variables. It is, in this implementation, regression in multiple dimensions.

Two MLR models were built. The first model required all inputs to be used explicitly; however, the weight on any given attribute may be zero. The resultant model will be of the standard regression form found in Equation 3.3.

$$y = \sum_{i=1}^n c_i x_i \quad (3.3)$$

where  $y$  is the predicted phase length, and  $c_i$  indicates the weighting constant associated with attribute  $x_i$ . The  $i$  subscript indicates the attribute, up to  $n$  attributes.

The second model used the R-squared value was to determine which inputs or combination of inputs (interactions) to include in the model. Fitting starts with that linear fit model (from Equation 3.3) and in a stepwise manner adds and removes terms, up to the maximum complication of terms (e.g. square, cubic, interactions, etc).

Table 3.2: Neural Networking neurons-per-layer sensitivity. All units are seconds.

	RMSE	$\sigma$	Computation Time
2 neurons per hidden layer	8.3	0.3	61
4 neurons per hidden layer	8.0	0.3	102
8 neurons per hidden layer	8.4	0.5	550
16 neurons per hidden layer	8.6	0.5	2300

### 3.7.4 Neural Networks

Neural Networks are computer simulations of neural pathways similar to that found in mammals. Neural networks have layers of neurons between the inputs and outputs, which create the so called network. The number of neurons and layers, input and feedback delays determines not only the computational complexity of the network, but also its ability to efficiently approximate more complex systems.

While the other methods described in this section are deterministic, a neural network requires a seed to start building the network. This means that, unless the seed characteristics are saved, the resulting neural network may perform differently based on different seeds. This seed based variance can be seen in Figure 3.2, where 10 seeds were run for each number of neurons to obtain an average RMSE and a standard deviation. In this Table, only a single layer of neurons was used; there were up to 4 input delays, and up to 4 feedback delays. No gains were observed beyond 4 neurons, and computational time rose drastically. It is important understand the implications of network complexity and the effects of seed selection.

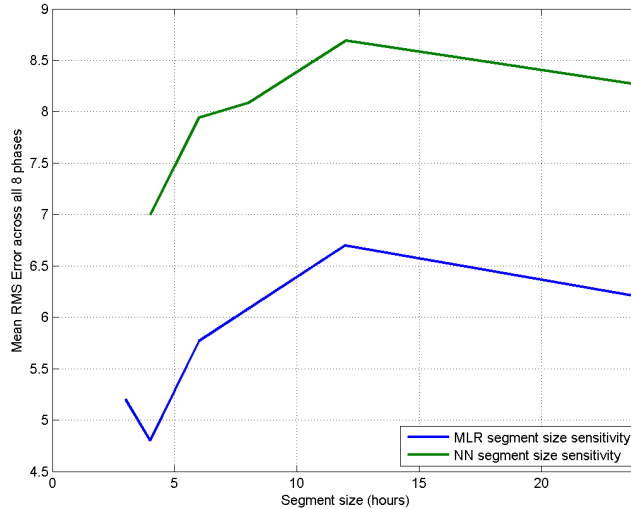


Figure 3.10: Method performance given varying amounts of data (from signal 0035 in Fremont, CA). Hours of data (for training, also for testing) given on X-axis, mean performance across all 8 phases given across Y-axis.

## 3.8 Discussion

### 3.8.1 Results

In Figure 3.10, multi-linear regression and neural networking training and test data sets were divided into smaller time segments, to determine if the amount of training data significantly affected method performance. It was initially thought that using training data from a narrow window (e.g. on-peak vs off-peak) would positively influence test results, however that was not found to be the case. While performance may be slightly altered by adjusting the amount of training data, regression clearly outperforms neural networking at all segment sizes up to the maximum tested, 24



hours. The lack of results for short segments (e.g. segments less than 4 hours) is a result of insufficient training data in the models leading to insufficient training or test matrix rank. It is interesting to note that for the NN models, the through phases perform worse as training data volume goes down, yet the turn phases perform better as the training data volume goes down; this may be a reflection of the relatively large variance in through phases.

Results presented here are for an example traffic signal in each test location (Fremont, California, Portland, Oregon and San Jose, California). In these tests, the attributes *elapsedPhaseLength*, *shortMean*, *longMean*, *waitTime* (for all 8 phases), and *vehicleCall* (for all 8 phases) were used as inputs in both training and testing the algorithms. These signals represent worst-case scenarios, with the most complicated control schemas, and the most phases.

The results found in Figure 3.11 reflect 24 hours of training and test data for each test site. Training data was recorded on a Monday and test data was recorded on a Tuesday. Multi-linear regression obtained the lowest root mean square error in Fremont, California. In Portland, Oregon, Multi-linear regression obtained the lowest root mean square error. In San Jose, California Transition probability modeling obtained the lowest root mean square error, with multi-linear regression performing second best. These results have lower error than the closest comparable results, found in [4]. When comparing the same prediction methods, in [4] it is not clear how that author arrived at the neural network design he did; results here show the process of arriving at the design used, and the change in error for variance from that design. In addition, different attributes have been used: the process of select-

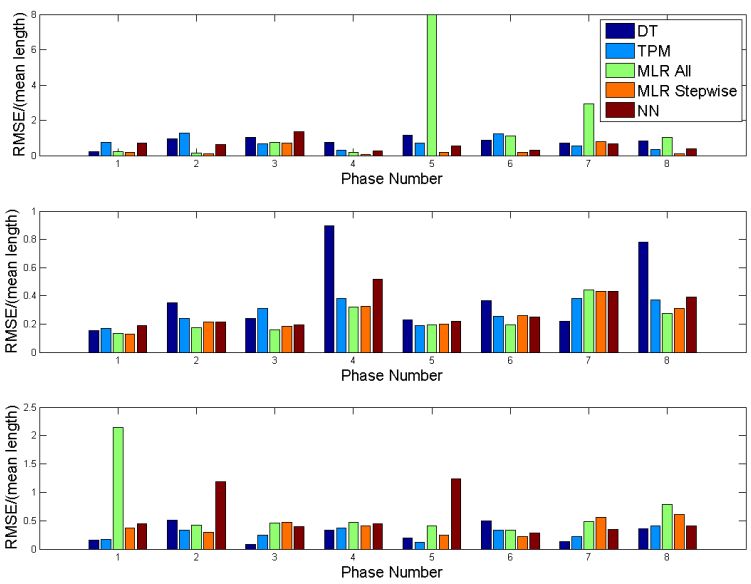


Figure 3.11: Root mean square error divided by mean phase length for test data from Fremont California, Portland Oregon, and San Jose California, with algorithms provided 19 attributes.

ing attributes here, using PCA, is described; no such process is described in [4], so the authors are not able to compare processes. Finally, different traffic signals have been evaluated. Differences in predictability exist even for signals with the same hardware,firmware and number of phases, due to installation parameters related to the controller operation and the specific intersection geometry. Still, the results here have significantly reduced error, even when utilizing a more pessimistic metric.

Both Linear Discriminant Analysis and Naive-Bayes Classifiers were used, and while in some situations they provided reasonable results, their dependence on full rank and positive covariance matrices makes their use in this implementation difficult, especially if small volumes of training data are used (e.g. < 12 hours). For this reason, they are not included here.

It is easy to get discouraged by a cursory review of Figure 3.11, because occasionally a specific method performs poorly. However, what should instead be taken away from this, is that an approach like bootstrap aggregating (bagging) of various methods will provide a more consistent level of error. The goal of showing the specific algorithm results in Figure 3.11 is to show the variance in methods, signals, and locations - and the successes with specific methods in certain situations. Future work could potentially evaluate aggregated results across a larger number of signals.

While the authors have not exhaustively examined all prediction techniques, or attributes, the goal of this manuscript was to determine the feasibility of predicting traffic signal timing information for use in vehicle systems. In determining for example whether an automatic start-stop system should be allowed to shut down the engine, two factors have the biggest impact on existing systems - the presence

or absence of an intersection, and the error. If a system is able to determine that the vehicle is approaching a traffic jam or a stop sign instead of an intersection, this is important. The second effect, that of prediction error once it is determined that the vehicle is approaching an intersection, is more application specific; an RMS prediction error of 3.6 seconds would be sufficiently accurate to significantly reduce the number of engine shutoffs immediately followed by engine startups caused when a vehicle approaches a signal about to change. In addition, this is why a level of confidence is provided, to allow the automatic engine start-stop system to determine if the level of confidence in the prediction is sufficiently high to prevent engine shut-off. Similarly, in determining whether the vehicle under braking should begin downshifting in the approach to an intersection, this level of error would be sufficiently accurate to determine whether to hold or downshift gears. However, for a velocity recommendation algorithm or a red light runner (collision prediction) system, where very high system accuracy is safety critical, it may prove insufficient. In addition, if a velocity recommendation system is able to provide a recommendation in some circumstances, and not in others, because the prediction confidence is low, or data is not available, this will provide a poor user experience. It is therefore recommended that these methods be used in internal algorithms, and not customer facing driver assistance systems.

### **3.8.2 Sources of Error**

Prediction error is significant, despite extensive analysis. We have found the error can be divided into two types: errors which are inherent to the system, and are

present in the raw data, and errors which are a direct or indirect result of our modeling methods.

Errors which are part of the system and present in recorded data are:

1. Arrival of new vehicles or pedestrians. When a prediction is made based on the available inputs, it may be correct at that moment. However, when a new vehicle or pedestrian arrives, it will influence the following phase lengths and any previous predictions which were previously correct may now be incorrect. This is impossible to avoid, and indicates that despite perfect data and perfect algorithms and implementation, it will never be possible to achieve 0 error.
2. Issues of missed data. In the Fremont raw data, it appears that data is occasionally lost. This could be because of transfer protocols used, other processes running on the host TMC computer, or dropped connections between the signal and the TMC computer or the TMC computer and our mirrored computer. This causes issues because if for example, the start of yellow or start of red is not recorded, this will result in a long green for that phase in that cycle, which is only present in recorded data and not present at the actual signal.
3. Discrepancies with documentation. While documentation like the Siemens Training Guide [40] indicate the effects of certain parameters, the observed effects appear to be in conflict with the descriptions found. For example, when a vehicle is in  $max_{green}$  mode, it is expected to last the duration of this parameter. In fact, the duration while the signal shows this  $max_{green}$  parameter contains significant variance.

4. Calculation of ground truth data. Ground truth data is calculated backwards in time, based on whether the active phases at the earlier timestep are the same. If so, active phase lengths are incremented by the time differential between records. This may introduce error to the system when the recording computer system loses connection with the traffic signal, and regains connection with the same phases active.
5. Imperfect attribute set. Because the authors are to some extent reverse-engineering the attributes going into the traffic signal controller algorithm without having access to the algorithm or the parameters set by the city, it is not possible to know if we have accounted for all possible influences. For example, the influence of conflicting vehicle calls has been included in this model, but the exact effect on current and future phase lengths is not known. In addition, the influence of ring order has not been evaluated or built into these models.

Errors which are specific to the methods we have used are:

1. Transition Probability Model. In Fremont, because data was recorded at approximately 1Hz, and traffic signals updated the server at approximately 1Hz, a small amount of aliasing was observed, whereby data was occasionally recorded twice and the next message was not recorded. The aliasing of data will affect the transition probability map slightly, but this effect was statistically insignificant. To avoid the issue completely, Portland and San Jose were recorded in an event-based manner. This has the potential draw-

back however, that diagnosing a scarcity of messages between the TMC and the mirroring/recording server is made more difficult as it could be either a dropped network connection or simply a lack of events (though this is solvable using communications requiring handshakes or message receipts, for example TCP connections).

2. Decision Tree. In this method, the previously discussed imperfect attribute set negatively impacts the performance of the decision tree directly. The goal of a decision tree is to recreate the decision making process of the deterministic traffic signal controller algorithm, yet because the inputs do not match exactly, its performance is not the same.
3. Multi-Linear Regression. Multi-linear regression does not allow for highly non-linear models. In addition, because our metric of RMSE is not the metric used directly (R-squared is used) to determine which terms are added or removed from the stepwise model, the stepwise algorithm performs differently than a standard linear fit.
4. Neural Networks. Neural Networks allow non-linearity, yet it did not perform consistently better than other algorithms. This, despite varying neurons per layer, varying layers, and varying inputs. This is a potential area for future work.

### **3.8.3 Discussion**

In this Chapter, we seek a predictive model of traffic signal phase and timing information such that in-vehicle driver assistance systems and powertrain control algorithms can make the vehicle both safer and more efficient. To that end, we begin with an evaluation of the data sets to the application of statistical predictive models. From there we build several machine learning models, to evaluate their relative applicability to the prediction of SPAT information. Finally, we discuss the many sources of error, both in the data itself and in the model building process, and evaluate the results.

## **3.9 Acknowledgement**

This work was performed in collaboration with, and supported by, BMW Group. The authors would also like to thank Dr. Zijun Wang for his contributions to the algorithms presented above.



# Chapter 4

## In-Vehicle Velocity

## Recommendation Architecture

## Evaluation

### 4.1 Disclosure

*This work was performed as a group consisting primarily of Andreas Winckler, myself, and a variety of BMW interns over several years. While I have played a significant role in the development of the system, in coding, in bug fixing, and in testing, it is not solely mine to take credit for.*

## 4.2 Abstract

The main contribution of this chapter is experimental verification of a system architecture for providing real time communication of individualized Traffic Signal Phase and Timing (SPaT) data, tailored to a specific vehicle for use in that vehicle's driver assistance systems. The role of data collected directly from Traffic Management Centers is explored; in addition, the collection and condensation of crowdsourced SPaT data is investigated as a complementary solution to providing SPaT data in situations where timing information is not available directly from a city's Traffic Management Center (TMC). Experiments with a number of drivers show that when data is available from a TMC, a 8.4% decrease in fuel usage is possible with the use of a velocity recommendation engine. When data from crowdsourced GPS traces is available, the system functions as intended; however, as a result of excessive traffic at the test location, drivers were not able to vary their speed according to the recommendation and a negligible difference in fuel usage has been observed.

## 4.3 Introduction

The process of improving transportation infrastructure is a balance between costs [6], and return on investment [41]. Recent technological advances may be shifting that balance [42]. It is now possible to tailor traffic signal communication to and from individual vehicles. The benefits of communication are significant. Prototype safety services from the Vehicle Safety Communications Consortium (Ford, GM, Honda,

Mercedes-Benz, Toyota) like the Smart Intersection and its related vehicular position communication technologies are able to warn a driver of other vehicles which are likely to run conflicting red lights at the upcoming intersection, potentially avoiding accidents [43, 44]. Similarly, with the communication of traffic signal information, it is possible to improve fuel economy through appropriate speed recommendations [11, 29, 19]. And while little is publicly available, the Audi Travolution project conceptually communicates traffic signal timing for in vehicle use, though it is not clear exactly what driver assistance systems would be available [23].

What these examples all lacked at the time of their writing was appropriate communication technologies. Many required Dedicated Short Range Communication technologies as part of their solution, and yet 14 years after the allocation of spectrum, this technology is present in neither cars nor infrastructure. There is significant cost associated with equipping the US' 330,000 traffic signals and all future vehicles with this technology [6].

However, major rollouts of 4G/LTE by wireless carriers, significantly increasing the bandwidth and decreasing the cost of broadband wireless internet, are opening new options. Similarly, major rollouts of IPv6 technologies, significantly increasing the number of uniquely addressable devices, creates new opportunities for communication infrastructures. The combination of these two sets of technologies now makes it feasible to send and receive Signal Phase and Timing (SPaT) information directly to and from individual vehicles.

In contrast to DSRC, what we are proposing is to use existing infrastructure and technologies. Some cities already collect and centralize SPaT information for

management purposes. Existing broadband communication architectures can then be used to transmit this data stream to servers for collection and analysis. In-place wireless technologies like 4G/LTE can then get the data from the servers to the vehicles. This system is implementable today and we are now testing it in prototype development vehicles.

Prior art reveals considerable work in the area of red light avoidance, and many conceptual solutions. However, the authors did not find public documentation of complete, implemented systems, or objective evaluations of their benefits. The big picture concept of using real time traffic signal information in speed advisory algorithms is not new and has been discussed in [11, 45, 12] and others. One of the most complete systems is that found in [13], wherein a system is developed based on cell phone cameras and ad-hoc wireless networks. While instrumental in shifting opinion on the feasibility of these types of systems, the proposed set of technologies have several drawbacks including a lack of extended communication range, inferior driver visibility as a result of devices attached to the windshield, and limited evaluation of the various kinds of traffic signals.

As an incremental development step, when real time but not future SPaT data is available, probabilistic techniques such as those discussed in [19, 28] may be one solution. A similar set of solutions is also offered in [30], with the primary difference being that the speed advisory algorithms depend more on work-energy algorithms and less on red light avoidance. All of these require some form of SPaT data, and a new method for gathering traffic signal data from numerous probe vehicles, the prerequisite step to providing the same to speed advisory algorithms, is discussed

in [36]. If historical SPaT data is available, [18, 4, 46] offer novel data mining techniques for predicting future SPaT information, and analysing the quality of said predictions. The authors seek to build on these works by offering a complete solution to providing individualized SPaT information directly to vehicles.

In this chapter, Section 4.4 describes the overall system architecture of how we get information to a vehicle. Section 4.5 describes where we get the information we are sending to the vehicle. Section 4.6 describes one potential use case for the information, once it has made it to the vehicle. Problems encountered in implementation are expounded upon in Section 4.8. And in Section 4.9 we review the effects of this driver assistance system, with different information sources, on vehicle fuel economy.

## 4.4 System Architecture

A system architecture is sought which is able to collect, analyze, and distribute to vehicles signal phase and timing data. The architecture must discriminate in sending specific intersections to specific vehicles, must scale to a significant number of intersections and vehicles in an efficient manner, and must cover various types of intersections and their respective technologies.

The following subsections will describe how we determine which information to send to which vehicles, and how we get that information.

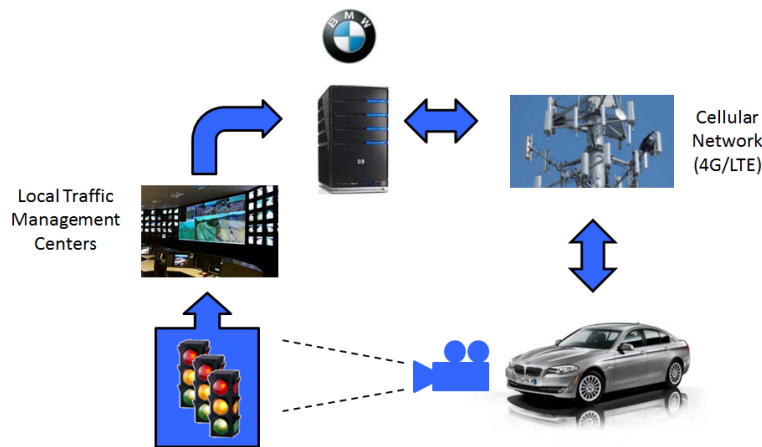


Figure 4.1: System overview of the path of data from a traffic signal to server to a vehicle. The path of data from the vehicle camera is described in more detail in Section 4.5 Subsection 4.5.3.

#### 4.4.1 Vehicle Subscriptions

In determining which traffic signals are relevant to a vehicle along a trip, a database of traffic signals is stored locally in the vehicle. For prototype development purposes, this initial database contains only those traffic signals located in the appropriate zip code. The vehicle also stores a database revision number, which is verified against the revision number on the server at vehicle startup. If a difference exists, either the differences between revisions, or in the case of corruption an entirely new database can be downloaded.

Within that database, in order to identify the upcoming traffic signal movement most relevant to the vehicle, we used a three step process.

1. In the first step, we conduct an initial scan through the database to determine traffic signals within a specified range of the vehicle (e.g. 3 miles).

2. In the second step, we use this list of traffic signals within range, and compare vectors of relevant traffic signal movement entry headings with current vehicle heading. This removes signals which are behind the vehicle.
3. In the third step, we select the closest signal from amongst the previous list. This is required because along a straight road we may have multiple signals on the same heading which are close matches for our current heading.

This three step process identifies and selects the upcoming relevant traffic signal movement. While this may lack potential benefits achievable with full future knowledge of all upcoming traffic signals along a route, it is an incremental step in that direction, and it is implementable today.

With the appropriate movement selected, the vehicle then sends the server a subscription request for the relevant phase. The server will respond immediately with the current status of that phase. Additionally, any time that traffic signal has an update (e.g. phase changes from green to yellow, placement of a pedestrian call, or a vehicle call), the appropriate information is forwarded to the vehicle.

After a vehicle has passed through an intersection, that intersection is no longer relevant, and the vehicle sends an unsubscribe message to the server indicating the vehicle no longer wishes to receive updates on that intersection. If another intersection is now relevant according to the 3 step process outlined above, that intersection will be subscribed to, and the process continues.

## 4.4.2 System Backend

Data from traffic signals intended for vehicle consumption is initially stored in system RAM on a scalable cloud-based server, allowing for fluctuations in number of connected intersections and number of connected vehicles requesting SPaT information (and the associated fluctuations in bandwidth, storage capacity, and performance).

Messages received on the backend indicate a change in the current status of a traffic signal and addition or removal of a connection to a traffic signal. If nothing has changed since the previously recorded status of a traffic signal, no message is sent, no record is made.

## 4.5 Data Sources

While traffic signals in some major cities are connected to traffic management centers, drivers will likely encounter many signals which are not connected to their city's traffic management center. If a driver assistance system is customer facing, drivers will not understand if their driver assistance system is only able to provide recommendations at certain intersections at specific times of the day. It is imperative to provide a reliable user experience; this requires broad coverage of traffic signals across most geographical areas drivers are likely to take their vehicle.

The majority of data used in this manuscript is received from TMCs. However, for those signals which are not connected, predictions of green and red switching times are used, based on crowdsourced data as described below in Subsections 4.5.2 and 4.5.3. Subsection 4.5.2 describes how to collect SPaT data and build predictions



from crowdsourced GPS traces. Subsection 4.5.3 describes how to collect SPaT data and build predictions from crowdsourced in-vehicle camera data. These two approaches complement the collection of data from traffic management centers, in order to provide consistent coverage across as large an area as possible.

From the driver’s perspective, there should be no difference between data collected directly from traffic management centers, data collected from crowdsourced GPS traces, and data collected from on-board cameras.

#### **4.5.1 Traffic Management Center**

Traffic management centers operate by polling connected traffic signals for current status information. In the cities used as test fields for the purposes of this manuscript, that polling occurs at approximately 1 Hz. This data is available to city engineers, and allows a traffic engineer to sit at her/his desk and verify that the signal is operating exactly as the engineer intended. This also provides an interface from which it is possible to extract the real-time data into another system, for example when updating to a new signal controller, or when connecting an outside system such as that described in this manuscript. Details of the technologies for providing city data vary from city to city. As this data is owned by the public, it is available to anyone that requests it, provided the city has the technical capabilities in place.

## 4.5.2 Crowdsourced Data via GPS Traces

For traffic signals not connected to traffic management centers and public infrastructure, another approach is necessary to provide traffic signal phase and timing data. Specifically, recent publications have made it clear that probe vehicle GPS traces could potentially be used to determine traffic signal timing.

The feasibility of generating SPaT information from crowdsourced probe vehicle GPS traces has been shown by our group in [36] and the overview is provided here:

1. Aggregate statistically significant number of vehicle traces/probes
2. Identify and filter data around traffic signal locations
3. Rebuild vehicle “most likely” velocity trajectory
4. Estimate phase and or phase change of signal
5. Build phase estimator from statistically significant number of observed phases/phase changes
6. Estimate future phase/phase changes

For testing and evaluation purposes, a public feed of bus location and velocity data from the city of San Francisco is being used to crowdsource the collection of traffic signal information. The feed is being provided by NextBus Incorporated [47] through eXtensible Markup Language (XML). The XML feed can be accessed using URLs with parameters specified in the query string.

A server continuously receives vehicle location updates and stores the resulting XML data in our SQL database. Another node of this server is dedicated to estimating the traffic signal phase and timing information including cycle length, phase length, offset, and signal schedule changes. More details of the algorithms can be found in [36].

In order to compare the estimates with real-time signal states at the intersections, a web server with a PHP interpreter was built. The web server connects to an SQL database to read the most updated crowdsourced start-of-green as well as the timing of the desired intersection.

With estimates of future phase and phase changes from the crowdsourced data in hand, it becomes a matter of sending a UDP message to the signal server every time a phase change is predicted. The signal server is therefore able to treat this data source exactly the same as data coming from a traffic management center.

### **4.5.3 Crowdsourced Data via On-Board Cameras**

The test vehicle contains a next-generation MobilEye camera and image processing system (EYE-Q) capable of detecting traffic signals. The image processing system provided by MobilEye reports recognized traffic signals on the CAN bus [48]. Our software then utilizes the traffic signal heading and distance from the camera host (ego) vehicle, and ego position and heading, to determine the approximate traffic signal location. This approximate traffic signal location is then compared with the list of known traffic signal locations. If a close match is found, the status of the detected traffic signal, along with a timestamp, are sent to and stored in the

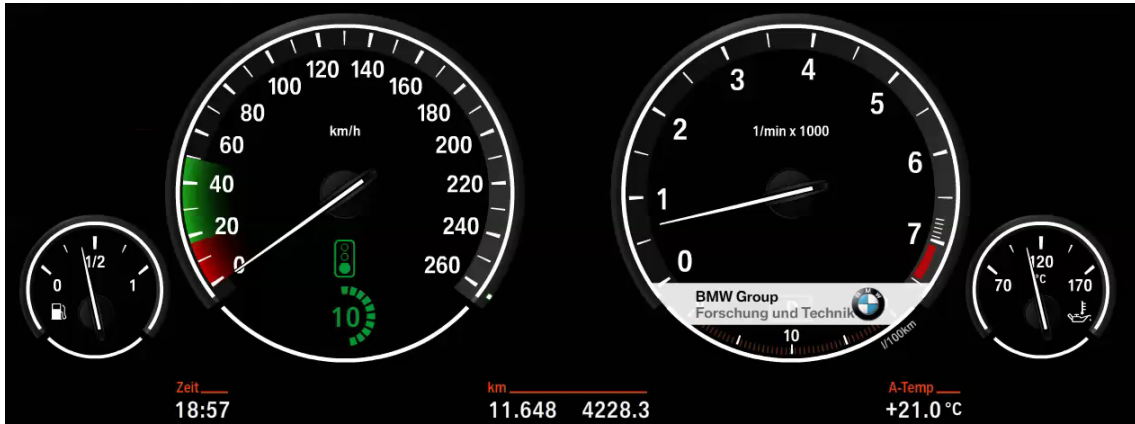


Figure 4.2: Driver’s dash display, including speed recommendation and countdown to green.

crowdsourcing back end. If no known traffic signal is nearby, a new database entry is created and the traffic signal location and status information is sent and stored in the crowdsourcing back end.

From the data collected by MobilEye cameras, we are building a database of traffic signal locations and traffic signal statuses. Other authors have examined the effects of penetration rates on fleet efficiency [18, 49]. And our group has shown it is possible to utilize even more sparse data to predict upcoming traffic signal information [36].

## 4.6 Speed Recommendation as Exemplary Use Case

While data of this nature has uses in safety applications like red-light-runner warnings, and use in efficiency applications like automatic motor start-stop, in this section we focus primarily on a speed recommendation engine. A velocity advisory algo-

rithm was devised which recommends the appropriate speed range to pass through the next upcoming traffic signal during the green phase, as described in Subsection 4.6.1. The appropriate speed recommendation is displayed to the driver as green and red zones on the speedometer as seen in Figure 4.2. In addition, when an upcoming phase change is known, a countdown will appear at the center of the speedometer indicating the remaining time to change of phase.

In Subsection 4.6.2, the velocity recommendation algorithms are evaluated at two test fields, in San Jose, California and in San Francisco, California.

#### 4.6.1 Velocity Advisory Algorithm

In general, the primary objective is to avoid stopping at a red light if at all feasible. Idling may require fuel, depending on the vehicle. Even in vehicles with automatic start-stop, the deceleration and associated acceleration back to cruising requires significant quantities of fuel. It is also important to note that an on-board camera ensures that speed recommendations calculated in this section are not made to drivers which could result in a collision; the recommendations are turned off when impeding vehicles are detected which are either going significantly slower or are stopped.

It is assumed that the approximate distance to the next traffic light(s) is known at each time and shown by  $d_i$ . The subscript  $i$  denotes the light number in a sequence of traffic lights, i.e.  $d_1$  is the approximate distance to the first upcoming light and  $d_2$  to the second light at each time. The light(s) update and broadcast their expected sequence of green and red times regularly. Suppose  $g_{ij}$  is start of the  $j^{th}$  green of

the  $i^{th}$  traffic light and  $r_{ij}$  is start of the  $j^{th}$  red of the  $i^{th}$  light. For example, when the timing of light number 1 is known, a recurring sequence

$$[g_{11}, r_{11}, g_{12}, r_{12}, g_{13}, \dots] = [40, 100, 150, 200, 240, \dots]$$

which implies the first traffic light is currently red, it will turn green in 40 seconds, red in 100 seconds, green again in 150 seconds, and so forth is possible. Figure 4.3 shows a schematic of the map formed at each time step based on the information received from the lights. Equipped vehicles can use the remaining distance to the next light(s) and the green and red sequence to set their target speed. This target speed (slope of each path) should be in the feasible range  $[v_{min}, v_{max}]$  where  $v_{min}$  is the road's minimum speed limit and  $v_{max}$  is the road's maximum speed limit. Other constraints, such as maintaining a safe distance to the vehicle in front, are handled separately.

The following steps determine the target speed at each step  $k$ :

1. For a vehicle to pass during the first green of the first light, its velocity should be in the interval  $[\frac{d_1}{r_{11}}, \frac{d_1}{g_{11}}]$ . This is only feasible if this interval has a set intersection with the feasible speed interval of  $[v_{min}, v_{max}]$ . If this set intersection is empty, passing through the first green without stopping at red is deemed infeasible. In that event, feasibility of passing during the next green interval is checked and the process is repeated until for some  $i^{th}$  interval  $[\frac{d_1}{r_{1i}}, \frac{d_1}{g_{1i}}]$  has a set intersection with  $[v_{min}, v_{max}]$ . This set intersection is mathematically characterized by:

$$\left[\frac{d_1}{r_{1i}}, \frac{d_1}{g_{1i}}\right] \cap [v_{\min}, v_{\max}] \quad (4.1)$$

and determines the range of speed that ensures passing the first light without having to stop at a red.

2. If passing without stop at the first light is determined to be feasible, the process in step 1 is repeated for the second traffic light by checking the set intersections

$$\left[\frac{d_2}{r_{2i}}, \frac{d_2}{g_{2i}}\right] \cap [v_{\min}, v_{\max}]$$

and picking the first non-empty one.

3. Next, the set intersection of the feasible range of speeds determined in step 1 and that of step 2 is calculated. A non-empty solution  $[v_{low}, v_{high}]$  indicates feasibility of passing the two lights without having to stop at a red by maintaining a constant speed. However an empty solution does not imply that stopping at red is necessarily required. It only means that passing the two consecutive lights with the same speed is not feasible. In that event, the vehicle can re-adjust its target speed after passing the first light to pass the green of the second light.
4. The process is continued by checking the next lights until a stop at red becomes unavoidable. The last feasible range  $[v_{low}, v_{high}]$  is an appropriate target velocity range. In this paper we set the speedometer to green for the range  $[v_{low}, v_{high}]$ , with the target of reducing fuel usage and trip time.

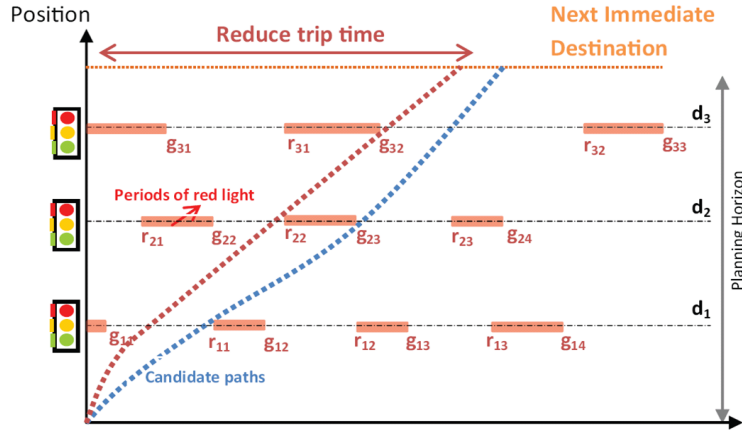


Figure 4.3: Schematics map of red lights distributed over space-time. The graphics shows how an informed vehicle passes two consecutive traffic intersection without having to stop at a red.

While this process describes what could happen in the case where timings for multiple signals are known, and known for some number of cycles in the future, *only the first signal and only the first cycle are used for speed recommendations in this manuscript*. This was done for two reasons. The technical reason was that future cycles are not necessarily deterministic in actuated signals. The human factor reason is that the idea of passing through a single signal is the easiest concept for drivers to understand.

Note that the velocity range is updated at each sampling time and therefore may change at each instant based on vehicle's position and the most recent information from the lights. This set of rules is not necessarily "optimal", but helps break down a fundamentally non-convex optimization problem to a simpler real-time implementable one.



## **In-Vehicle Camera Integration with Velocity Advisory Algorithm**

The presence of traffic at an intersection could potentially affect how a vehicle seeking to pass unimpeded through the intersection should behave. For example, the absence of any vehicles in the ego vehicle's lane waiting at a red light suggests that the connected vehicle could arrive at the stop bar shortly after the light turned green. However, the presence of another vehicle in the ego vehicle's lane would alter the ego vehicle's speed profile, depending on the rate of acceleration of the other vehicle and other factors. Similarly, the presence of another moving vehicle in the ego vehicle's lane as the ego vehicle approaches a traffic signal could also alter the ego vehicle's speed profile (and appropriate speed recommendations).

The presence (or absence) of other vehicles on the road can be detected by the MobilEye camera, and updates about obstructing vehicle position, speed, and other information are sent by the camera over the CAN bus. This allows the speed recommendation engine to provide recommendations that take into account other vehicles on the road.

For the results presented in this manuscript, the camera was used for detection of vehicles travelling significantly slower than the ego vehicle, and for detection of vehicles stopped at intersections. In both cases, if it was unsafe to display a speed recommendation because the ego vehicle could potentially impact the detected vehicles by following a speed within the recommended range, the recommendation was turned off.

## 4.6.2 Fuel Economy Evaluations

The above algorithm was implemented in a test vehicle, and the following subsections describe evaluation of that algorithm in real world test environments. Two test environments have been selected - the cities of San Jose, California and San Francisco, California. They have been selected to test two different data sources. In San Jose, the traffic signal data will be coming directly from a traffic management center, and is *assumed* to be correct. This may not be true in 100% of cases, but is sufficient for testing purposes. In San Francisco, the traffic signal data will be coming from predictions based on crowdsourced GPS traces. Fuel economy is used as a metric to determine the accuracy of the data and efficacy of the algorithms, in the presence of actuated signals, traffic, pedestrians, buses, and other factors.

### Testfield San Jose, California

A test field has been set up in San Jose, California. Traffic signal data from approximately 800 traffic signals are available to the vehicle in real time. Approximately 150 status updates are recorded each second. Data from the traffic signals is first collected by a traffic management center, and then forwarded to a BMW cloud based server for analysis, storage, and re-distribution to appropriately subscribed vehicles.

A map of the test route can be found in Figure 4.4. This route was selected to evaluate the feasibility of the system in an urban environment; the test route crosses 2 sets of train tracks, passes a bus stop, a fire station, city hall, a superior court, and a construction zone. Pedestrians are present at many intersections, light to medium traffic was observed throughout all testing. The test route contains 3 right turns at

intersections, 3 left turns at intersections, and 8 intersections where the test vehicle is able to pass through without turning. Speed recommendations were available at 10 out of the 14 intersections; please see Figure 4.4 for further information about which intersections had recommendations available.

The test vehicle is a modified 2011 BMW 535i. The vehicle has additional test equipment, including supplemental power electronics to support multiple on-board computers for testing purposes. The vehicle has an automatic transmission. Automatic engine start-stop was not enabled in this development vehicle.

A total of 14 drivers were used. Drivers were asked to obey all road laws. Each driver was given approximately 20 minutes and 15 miles to familiarize themselves with the development vehicle. Initially, the authors gave turn-by-turn directions to drivers to ensure that they understood the test route. The drivers were then given one unrecorded lap of the test route, to ensure they understood where to go. Drivers were asked to drive for approximately 1 hour around the test loop with the speed recommendation and countdown systems *disabled*. The drivers were then asked to repeat driving the same route for approximately 1 hour, this time with the speed recommendation and countdown systems *enabled*. Drives occurred primarily from 9-11AM and from 1-3PM; every weekday was used, and no testing was performed on weekends. Drivers ranged from 18-40 years old. The fuel economy of each driver was recorded and can be found in Table 4.1.

The authors noted that in some instances, test drivers were not following the requested test procedures. Therefore, a second set of drivers were asked to perform as above, except the test time was shortened to a 30 minute session with the system

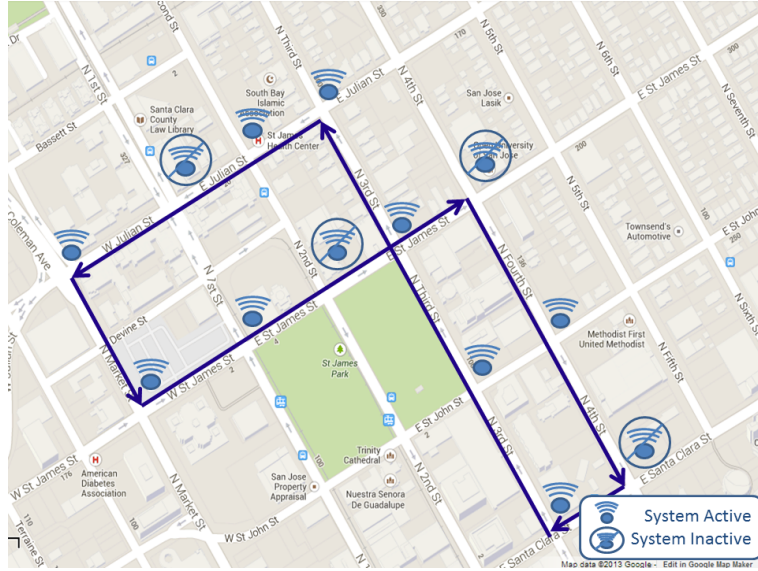


Figure 4.4: Fuel economy evaluation route in San Jose, California.

Table 4.1: Field testing in San Jose, California. Drivers were *not* given specific instructions to follow speed recommendations.

Driver #	System Inactive	System Active
Driver 1	11.4 MPG	13.0 MPG
Driver 2	12.8 MPG	13.4 MPG
Driver 3	14.3 MPG	13.2 MPG
Driver 4	12.3 MPG	14.3 MPG
Driver 5	14.2 MPG	13.3 MPG
Driver 6	13.2 MPG	12.6 MPG
Driver 7	12.0 MPG	11.4 MPG
Driver 8	12.9 MPG	12.4 MPG

Table 4.2: Field testing in San Jose, California. Drivers *were* given specific instructions to follow speed recommendations.

Driver #	System Inactive	System Active
Driver 1	13.5 MPG	14.4 MPG
Driver 2	12.7 MPG	13.5 MPG
Driver 3	13.2 MPG	15.9 MPG
Driver 4	10.9 MPG	11.2 MPG

off and a 30 minute session with the system on. Drivers were explicitly asked to follow speed recommendations, safety permitting. The results from those drives are found in Table 4.2.

With two drivers, data corruption lead to insufficient data to make analysis significant (2 or fewer recorded laps without issues). In one case, the test was cut short for safety reasons. In the other case the driver was unable to successfully follow the test route, despite the authors providing verbal directions.

### **Testfield San Francisco, California**

A test field has been set up in San Francisco, California. Data from buses driving public transportation routes in the city are used to estimate traffic signal switching times, as described in [36]. Data from the traffic signals is first collected by a crowdsourcing server, and then forwarded to a BMW cloud based server for analysis, storage, and re-distribution to appropriately subscribed vehicles.

Because the test route along which information is available, namely Van Ness Avenue, is short (approximately 1.1km), fuel efficiency along the test route is highly dependant on avoiding red lights. In addition, in this test we ran only 4 laps for

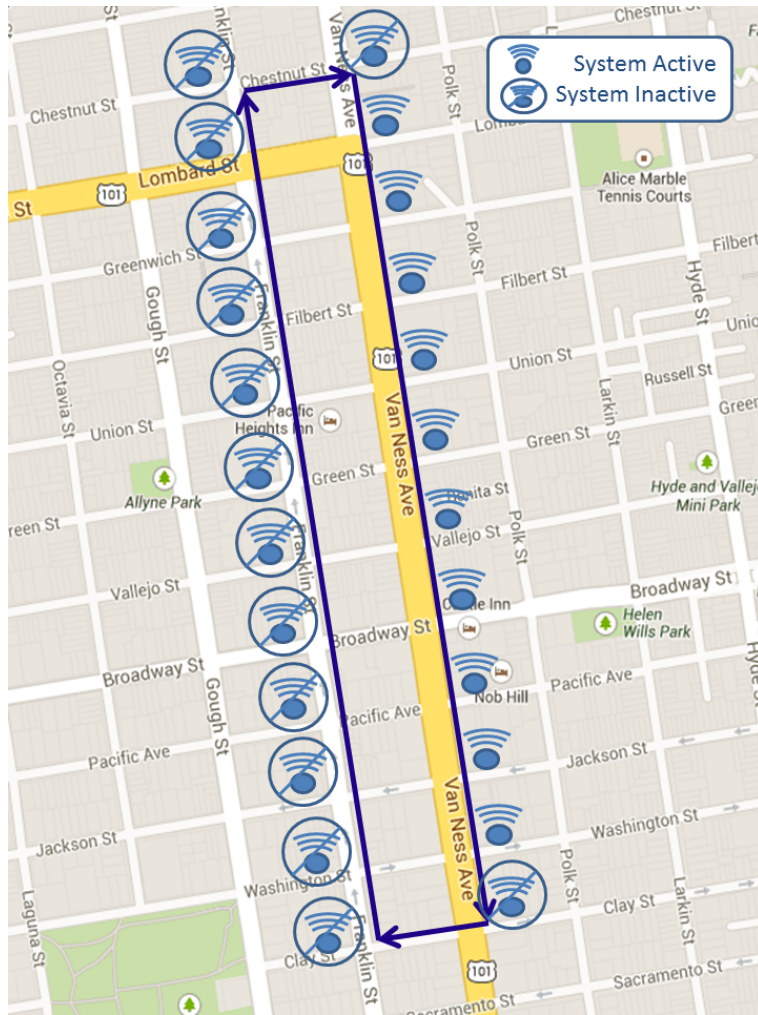


Figure 4.5: Fuel economy evaluation route in San Francisco, California.

Table 4.3: Field testing in San Francisco, California. Fuel economies in miles per gallon (mpg).

	Fuel Economy
Speed Recommendation Inactive	11.2
Speed Recommendation Active	10.8

each system state (off and on). Further complicating matters is the timing of these signals - some signals have red lengths of approximately 60 seconds. In this test case, though the inactive system fuel economy is slightly lower than when the system is active, this is likely the result of having stopped on one lap when the system was on for a nearly 50 second long red. This aberration skews the results significantly. Finally, traffic was moderate to heavy during the test phase, allowing the driver little freedom to vary his speed in accordance with the velocity recommendation algorithm. A more significant number of laps would be needed to conclusively evaluate whether the system is effective or not. This was not possible at this time due to the unavailability of both test drivers and the test vehicle.

While the fuel economy results are inconclusive, this did allow us to verify the prediction quality of the bus-crowdsourced phase change information. The on-board camera system recorded timestamps, locational data, and phase change data for when traffic signals changed. This data was then compared with the predicted results. The RMS difference (error) between the predicted phase changes and the camera-recorded phase changes, observed at 8 of the 10 intersections, over 21 phase changes, was 4.1 seconds. Both the prediction algorithm and the vehicle use Network Time Protocols to connect to time-keeping servers to verify that in this calculated error we are not including computer system clock time error. In addition to the 21 phase changes recorded and used in this calculation, 2 more phase changes were observed by the camera system, which did not align with predicted phase change intervals; these may be the camera system observing the wrong phase at an intersection, e.g. a left turn signal is visible to the camera but the through lane signal is

blocked from view by another vehicle.

## 4.7 iPhone Implementation

A rudimentary version of this has been successfully implemented in downtown Greenville, South Carolina on an iPhone 3GS. A screenshot of the application that was written is shown in Figure 4.6. Prior to implementation, it was unclear whether the 3G cellular data network would have low enough latency to retrieve the database information with sufficient time to make the velocity calculations as a vehicle was moving. Clock synchronization remained the largest issue; Verizon iPhones, ATT iPhones, traffic signal master controllers, and the National Institute of Standards and Technology disagree significantly on what time it is. A potential solution has been implemented in the iPhone application for testing purposes, and a human driver was able to successfully avoid red traffic signals when disturbances such as other vehicles or pedestrian traffic were low. This solution has many drawbacks by comparison to in-vehicle algorithms, including significantly reduced accuracy of positioning information, the previously described timing issues, limited computational power, and driver distraction complications related to having a separate screen instead of in-dash implementation. However, it was an important early evaluation of the feasibility of providing real time information, and provided initial feedback regarding future driver human-machine-interface issues and potential solutions.



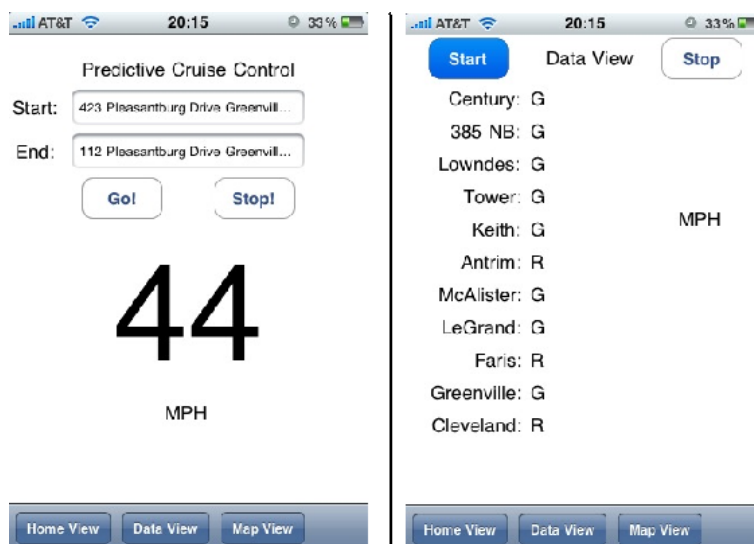


Figure 4.6: Screenshots of our iPhone application that uses SPAT information to calculate the speed to reduce idling at reds. On the left: dynamically changing optimal speed. On the right: instantaneous state of upcoming lights on Pleasantburg Drive in Greenville, SC.

## 4.8 Problems Encountered

In evaluating the collection, analysis, distribution and use of traffic signal phase and timing data as part of a speed recommendation fuel efficiency algorithm, a variety of problems were encountered. Issues ranged from technical errors to driver acceptance and attention issues.

Five technical issues were encountered.

1. Network connectivity. The 4G/LTE system occasionally dropped out for periods of 1-2 minutes as it lost signal and re-established connection. This means that drivers were not able to receive traffic signal updates during those periods.
2. Simple traffic signal selection algorithm. Occasionally while driving on an overpass, the system will mistakenly select the signal which is under the overpass, which is understandable given the role GPS position and heading attributes play in the algorithm.
3. The use of uni-directional UDP for many interfaces means that it is not possible to verify whether all data has been received, meaning the driver may be approaching a signal which has changed phase but the data was lost in transmission to the vehicle. In the case of a prototype system, and specifically in a system where overall network latency is a topic of research, UDP will generally outperform TCP. Additionally, in the case of information coming from cities to our servers, while a firewall is employed by the city, there is concern over interference from nefarious users. A unidirectional communication channel is

one piece of the security profile which ensures the continued safe operation of the traffic signal infrastructure. However, if this system is being used for a safety-focused driver assistance system such as a red-light-runner predictor, the communication technologies could change to ensure the vehicle's receipt of all safety messages.

4. Obtaining an accurate positional location of the vehicle. The GPS position available on the CAN bus is already map-matched, meaning that the raw GPS position, along with a basic Kalman filter, yields a GPS position which has been corrected to be on the internally stored map of roads. This may be different than the raw GPS position obtained directly from the sensor. This causes issues if other locational data, like traffic signal stop bars, is generated using non-map-matched data. However, this is merely a result of the specific test vehicle used, and is not a conceptual problem with the project proposed in this manuscript.
5. Camera systems and algorithms. Because the camera is a single lens, distance measurements become error prone. The vector based image recognition algorithm appears to need sufficient vehicle speed (and therefore difference between captured frames) to accurately estimate distance to objects identified by the image recognition system (like impeding vehicles, and traffic signals). This would suggest that a stereo system (e.g. two camera lenses, with significant horizontal separation between them) would increase the accuracy of the system.

The compounding effect of the GPS and camera errors, starting with the vehicle position and heading obtained from the GPS system/inertial positioning system, in combination with the error induced by the image recognition system in determining the distance and heading to the perceived traffic signal, causes some issues in identifying and correlating the perceived signal with the database of signals. The total combined error at the time of writing can come close to 30 meters, which may actually cause incorrect identification of the signal.

In addition to technical errors, at least three human interface issues were encountered.

1. Displaying speed recommendations. Initially a driver was given a specific speed to hit - a single number (derived from the target to pass through as many lights as possible). Drivers found this too difficult to follow, especially as the number changed frequently, as vehicle speed, signal timing changes, and traffic influenced matters. This was changed to the green/red areas on the speedometer (as seen in Figure 4.2), which while conceptually less efficient because they only relate to the upcoming signal and not a series of signals, actually can be followed by drivers.
2. Driver use of the system. Some drivers forgot about the system and did not pay attention. Some drivers got bored and started accelerating excessively, simply enjoying access to a powerful vehicle, with little regard for the system indicating an upcoming red signal.
3. Driver compliance with road laws. The system only displays recommendations

when the driver is following road laws. However, some drivers may try to exceed the speed limit if they are able to surmise from the provided information that they will not be able to pass the current intersection within the speed limits and recommendations provided by the system.

## 4.9 Conclusions

This chapter presented a vehicle architecture for providing real time traffic signal data to a vehicle for use in a speed recommendation algorithm.

In the first case study, performed in San Jose, California, exact traffic signal information was utilized to provide speed recommendations at 10 out of 14 traffic signals along the test route. The recommendation system resulted in a 0.8% average decrease in fuel consumption across the uniformed test drivers. The recommendation system resulted in a 8.5% average decrease in fuel consumption across the informed test drivers. These improvements are possible even in an urban setting, with traffic, buses, trains, pedestrians, and construction. Clearly, testing methodology and driver understanding plays a significant role in the results.

In the second case study, performed in San Francisco, California, estimated traffic signal information was used to provide speed recommendations at 10 traffic signals along a very short test route, only 1.1km long. This did not result in a statistically significant fuel economy difference, primarily because traffic was too dense. However, it did provide an opportunity to evaluate the quality of data coming from the crowdsourced data algorithms. Predicted phase timing compared to camera-

recorded ground truth data indicated an RMS difference (error) in prediction of approximately 4.1 seconds. With the two outliers excepted, the average error was 1.9 seconds.

Many effects including time of day, driving style, and the use of automatic motor start-stop were not studied in this manuscript and remain topics for future work. Additional future work will focus on the network wide effect of informed vehicles on uninformed vehicles, penetration rates, and fleet efficiency.

## **4.10 Acknowledgement**

This work was performed in collaboration with, and supported by, BMW Group. The BMW interns and employees who spent long hours driving the test route were invaluable to the work presented here.

# Chapter 5

## Summary of Contributions and Proposed Future Work

### 5.1 Review of Results

In Chapter 2, we statistically evaluated velocity planning algorithms which minimize idling time behind red lights based on probabilistic traffic signal timing models that we proposed. Three cases were evaluated - vehicles with no information about upcoming traffic signals, vehicles with real-time information, and vehicles with full and exact future knowledge of traffic signal timings. Drivers today fit into the first case - the least efficient. Drivers of the future may fit into the third and most efficient case, if infrastructure and technologies develop to provide this information. The middle case is feasible today, and obtains much of the potential benefit obtainable via knowledge of upcoming traffic signal timing.

Real time knowledge with probabilistic models where the driver encountered fixed-time lights yielded an optimistic 61% increase in a motivating case study, and a 16% increase in average fuel economy across 1000 multi-signal simulations of fixed time signals. The same models in combination with real time information yielded a 6% increase in fuel economy for actuated signals. These reflect technologies which could be feasibly implemented with little or no infrastructure changes and with only software updates to current production vehicles.

In Chapter 3, we sought a predictive model of traffic signal phase and timing information such that in-vehicle driver assistance systems and powertrain control algorithms would make the vehicle both safer and more efficient. To that end, we began with an evaluation of the data sets to the application of statistical predictive models. From there we built several machine learning models, to evaluate their relative applicability to the prediction of SPAT information. Finally, we discussed the many sources of error, both in the data itself and in the model building process, and evaluated the results.

In Chapter 4, we presented a vehicle architecture for providing real time traffic signal data to a vehicle for use in a speed recommendation algorithm.

In the first case study, performed in San Jose, California, exact traffic signal information was utilized to provide speed recommendations at 10 out of 14 traffic signals along the test route. The recommendation system resulted in a 0.8% average decrease in fuel consumption across the uniformed test drivers. The recommendation system resulted in a 8.5% average decrease in fuel consumption across the informed test drivers. These improvements are possible even in an urban setting, with traffic,



buses, trains, pedestrians, and construction. Clearly, testing methodology and driver understanding plays a significant role in the results.

In the second case study, performed in San Francisco, California, estimated traffic signal information was used to provide speed recommendations at 10 traffic signals along a very short test route, only 1.1km long. This did not result in a statistically significant fuel economy difference, primarily because traffic was too dense. However, it did provide an opportunity to evaluate the quality of data coming from the crowdsourced data algorithms. Predicted phase timing compared to camera-recorded ground truth data indicated an RMS difference (error) in prediction of approximately 4.1 seconds.

## 5.2 Novel Contributions

This dissertation contributes a globally optimal velocity planning algorithm, at least three techniques to provide SPAT information to said algorithm, and an evaluation of the algorithm and the SPAT information via empirical road tests.

The new approach in our globally optimal velocity planning algorithm provides the big picture for this dissertation - the Dynamic Program is the starting point of this project. However, the DP requires SPAT information in order to make velocity recommendations. To this end, we started with probabilistic methods, and moved from there. Multiple data sources have been explored, to improve the quality of information going into the velocity planning algorithm. Because of the raw volume of data, processing must be performed on a cloud-based service and the processed information made accessible to any device or connected vehicle. Such real-time information of current and future state of traffic lights does not exist today. New vehicle functions that rely on prediction of traffic signal state can be deployed starting today, locally and gradually, and do not require major involvement of government or a paradigm shift by auto-makers. By relying mostly on software, information, and fast cellular and Wi-Fi networks that exist today, traffic flow and safety can be improved and there can be dramatic reduction in CO<sub>2</sub> emissions and total national fuel use with direct societal and economic impacts [50, 51, 14, 52, 53, 54].

As described in Chapter 2,

- a globally optimal velocity planning algorithm
- a novel probabilistic approach to prediction of traffic signal phase and timing information
- a new application of Monte-Carlo simulations in the evaluation of velocity planning algorithms.

As described in Chapter 3,

- new utilization of two existing data mining methods in the analysis of historical traffic signal phase and timing information for prediction
- a new technique for evaluation of mined historical information.

As described in Chapter 4,

- a unique cloud-based database which is able to distribute just-in-time traffic signal phase and timing information for the relevant signals to the relevant drivers
- a new in-vehicle cruise control algorithm
- a new iPhone application for optimal velocity planning.

Experimental implementation of an in-vehicle algorithm as described in Chapter 4 validates the work in preceding chapters. These implementations demonstrate the full breadth of the project, from prediction to velocity planning, to in-vehicle implementations, to a final fuel economy gain.

## 5.3 Takeaways

What is at first glance a fairly simple question, upon second look is a rabbit hole of issues. The concept of getting traffic signal information into vehicles is an assumption in many works, yet no one has made public a viable, complete, system. The process of obtaining traffic signal data, which is owned by the public, turns into both a technical and bureaucratic nightmare, as paper records are often the most up-to-date records available and digitizing the information a tedious process. Cities may have traffic signals from the 1940s only a block away from brand new traffic signals, making the process of building a homogeneous phase and timing database increasingly difficult. Cities are strapped for funding, and the nature of technology is such that the replacement of every traffic signal controller in a city with a standard controller, firmware, software, and programming is unlikely. Every city engineer that touched the signal controller appears to have differing preferences as to how to set up the controller logic, and there are a wide range of intersection geometries with varying temporal traffic patterns. Signal controller manufacturers are themselves unsure as to the effects of various timing parameters during transition phases between timing plans. These combine to make a difficult set of prediction and implementation problems.

The biggest takeaway to me, is that a dispersed, passive measurement system is the most promising avenue for collecting traffic signal information for dissemination to connected vehicles. This has benefits including network stability, widely dispersed data collection costs, and enormous system flexibility. The greatest concentration

of sensors would match where data is needed the most.

## 5.4 Dissemination of Results

- Initial results from Chapter 2, including the case studies, were submitted and accepted to the 2012 American Controls Conference in Montreal Canada and were presented in June 2012.
- Results from Monte-Carlo simulations and traffic-adaptive signals have been accepted by IEEE Transaction on Intelligent Transportation Systems.
- Results from Chapter 3, including many of the data mining techniques, are ready to be submitted to the Journal of Transportation Engineering.
- Results from experimental implementation, as discussed in Chapter 4, are ready to be submitted to the IEEE Transactions on Vehicular Technology.
- An abbreviated version of the experimental validation will be sent to the IEEE ITSC 2014.

## 5.5 Future Work

While the preceding work has been successful, there is always a next step in research.

Data is not available from traffic management centers in very many cities. The infrastructure exists, but many cities have yet to consider the potential positive effects of making such data available as a public stream. This points future research towards the crowdsourcing methods described briefly in Chapter 4, and in [36]. This concept is just at the beginning, and has huge potential for future vehicle efficiency and safety applications.

The field of computer science, specifically the sub areas of data mining and machine learning, is experiencing its time in the spotlight and associated rapid growth. New methods, improvements to existing methods, and efficiency gains to existing methods, all portend future successes and increased accuracy and promise of this as a prediction technique. Additional time spent considering new and innovative attributes may yield increased accuracy; similarly, an improved understanding of traffic signal programming, and/or access to actual traffic signal code, may improve accuracy. Future work should look at hundreds or thousands of traffic signals, with future improvements in computational power, to evaluate machine learning methods across entire cities or multiple cities.

In-vehicle testing provided only a proof-of-concept application. Though well polished, it is not ready for mass production. As just mentioned, data is not available in all cities, and consumers would expect the system to work everywhere. While superficially trivial, there is an interesting research question regarding driver intent

when making turns. Steering wheel angle, blinker status, and vehicle position in intersection proved insufficient to predict driver intent - drivers often leave blinkers on when they don't necessarily intend to turn, or don't use blinkers at all. This is an important problem to solve before a system showing velocity recommendations or phase change countdowns is able to be shown to customers. It would not be safe to show the information for the wrong phase.

The communication infrastructure is an important step towards providing information to semi- or fully- autonomous vehicles. This has the potential for influencing efficiency algorithms like automatic motor start-stop, red-light-runner warning (collision prediction), and gear selection maps. But those are the tip of the iceberg. A vehicle which is able to react to perceived traffic signals, as in existing Google and other development vehicles, is the next step. What is shown in Chapter 4 would be one piece of the infrastructure in this next step - a vehicle which drives itself autonomously through more green signals than its human-driven alternative, with efficiency and safety benefits.

Finally, the most significant area this dissertation affects is route planning. Existing proprietary route planning algorithms often use node-link models where traffic signal timing is taken into account only via the effect it has on probe vehicle delay on the relevant link. Real time traffic signal information, and prediction models, has the potential to significantly influence route recommendation algorithms, like Google Maps and in-vehicle systems like Navteq. Route planning, travel time prediction, and traffic load density balancing are expected to be most affected in urban areas because of the high traffic signal density.



# Bibliography

- [1] G. Mahler and A. Vahidi. Reducing idling at red lights based on probabilistic prediction of traffic signal timings. In *American Control Conference*, Montreal, Canada, June 2012. <http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3648/4094>.
- [2] G. Mahler. Real-time traffic signal data driver simulation. Web. <http://www.youtube.com/watch?v=STZiyzi0k5M>.
- [3] Federal Highway Administration. Traffic signal timing manual intersection phase numbering schema. [http://ops.fhwa.dot.gov/publications/fhwahop08024/images/4\\_3web.png](http://ops.fhwa.dot.gov/publications/fhwahop08024/images/4_3web.png).
- [4] V. Khakhutskyy. Signal phase and timing prediction for intelligent transportation systems. Master's thesis, Der Technischen Universitat Munchen, 2011. [http://valonsoftware.com/resources/masters\\_thesis.pdf](http://valonsoftware.com/resources/masters_thesis.pdf).
- [5] National Science Foundation. Nsf Leads Federal Efforts in Big Data. [http://www.nsf.gov/news/news\\_summ.jsp?cntn\\_id=123607&org=NSF&from=news](http://www.nsf.gov/news/news_summ.jsp?cntn_id=123607&org=NSF&from=news).
- [6] National Traffic Signal Report Card. Technical report, National Transportation Operations Coalition, 2007. <http://www.ite.org/REPORTCARD/>.
- [7] Intelligent transportation systems for traffic signal control: Deployment benefits and lessons learned. Technical report, U.S. Department of Transportation. [http://ntl.bts.gov/lib/jpodocs/brochure/14321\\_files/a1019-tsc\\_digital\\_n3.pdf](http://ntl.bts.gov/lib/jpodocs/brochure/14321_files/a1019-tsc_digital_n3.pdf).
- [8] Traffic Congestion and Urban Mobility. Web. [http://tti.tamu.edu/infofor/media/topics/congestion\\_mobility.htm](http://tti.tamu.edu/infofor/media/topics/congestion_mobility.htm).

- [9] S. M. Chin, O. Franzese, D. L. Greene, H. L. Hwan, and R. C. Gibson. Temporary losses of highway capacity and impacts on performance: Phase 2. Technical report, U.S. Department of Transportation, Oak Ridge National Laboratory, 2004. [http://www-cta.ornl.gov/cta/Publications/Reports/ORNL\\_TM\\_2004\\_209.pdf](http://www-cta.ornl.gov/cta/Publications/Reports/ORNL_TM_2004_209.pdf).
- [10] B. Asadi and A. Vahidi. Predictive use of traffic signal state for fuel saving. In *Proceedings of 12th IFAC Symposium on Control in Transportation Systems*, Redondo Beach, CA, September 2009.
- [11] B. Asadi and A. Vahidi. Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time. *IEEE Transactions on Control Systems Technology*, 19:707–714, 2010.
- [12] J. Apple, P. Chang, A. Clauson, H. Dixon, H. Fakhoury, M. Ginsberg, E. Keenan, A. Leighton, K. Scavezze, and B. Smith. Green Driver: AI in a microcosm. In *AAAI Conference on Artificial Intelligence*, San Francisco, CA, August 2011. <http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3648/4094>.
- [13] E. Koukoumidis, L.-S. Peh, and M. Martonosi. Signalguru: Leveraging mobile phones for collaborative traffic signal schedule advisory. In *9th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, Bethesda, Maryland, June 2011.
- [14] P. Schuricht and B. Baker. Efficiency-increasing driver assistance at signalized intersections using predictive traffic state estimation. In *Proceeding of 14th International IEEE Conference on Intelligent Transportation Systems*, 2011.
- [15] S. Mandava, K. Boriboonsomsin, and M. Barth. Arterial velocity planning based on traffic signal information under light traffic conditions. In *Proceeding of 12th International IEEE Conference on Intelligent Transportation Systems*, St. Louis, MO, October 2009.
- [16] M. Sivak. Effects of vehicle fuel economy, distance travelled, and vehicle load on the amount of fuel used for personal transportation in the U.S.: 1970-2010. Technical Report UMTRI-2013-10, University of Michigan Transportation Research Institute, February 2013.

- [17] J. Wollaeger, S. Kumar, S. Onori, D. Filev, U. Ozguner, G. Rizzoni, and S. Di Cairano. Cloud-computing based velocity profile generation for minimum fuel consumption: A dynamic programming based solution. In *American Control Conference*, Montreal, Canada, June 2012.
- [18] M. Kerper, C. Wewetzer, A. Sasse, and M. Mauve. Learning traffic light phase schedules from velocity profiles in the cloud. In *5th IFIP International Conference on New Technologies, Mobility and Security*, Istanbul, Turkey, May 2012.
- [19] G. Mahler and A. Vahidi. Reducing idling at red lights based on probabilistic prediction of traffic signal timings. In *American Control Conference*, Montreal, Canada, June 2012.
- [20] H. Rakha, K. Ahn, K. Moran, B. Saerens, and E. Van Den Bulck. Virginia Tech Comprehensive Power-Based Fuel Consumption Model: Model Development and Testing. *Transportation Research Part D: Transport and Environment*, 16(7):492–503, October 2011.
- [21] Donald E. Kirk. *Optimal Control Theory: An Introduction*. Dover Publications, 2004.
- [22] Federal Highway Administration. Investigating advanced traffic signal control. <http://www.fhwa.dot.gov/advancedresearch/pubs/11044/11044.pdf>.
- [23] Audi. Audi travolution: efficiently through the city. <http://www.audiusanews.com/newsrelease.do?id=1812>.
- [24] USDOT. Connected vehicle: Vehicle-to-vehicle and vehicle-to-infrastructure technology test bed. [http://www.its.dot.gov/connected\\_vehicle/technology\\_testbed2.htm](http://www.its.dot.gov/connected_vehicle/technology_testbed2.htm).
- [25] C. Day and D. Bullock. Computational efficiency of alternative arterial offset optimization algorithms. *Transportation Research Record: Journal of the Transportation Research Board*, 2259:37–47, 2011.
- [26] M. Maile, F. Ahmed-Zaid, S. Bai, L. Caminiti, P. Mudalige, M. Peredo, and Z. Popovic. Objective testing of a cooperative intersection collision avoidance system for traffic signal and stop sign violation. In *2011 Intelligent Vehicles Symposium*, Baden-Baden, Germany, June 2011.

- [27] X. Yang, J. Liu, N.F. Vaidya, and F. Zhao. A vehicle-to-vehicle communication protocol for cooperative collision warning. In *Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, Boston, Massachusetts, August 2004.
- [28] G. Mahler and A. Vahidi. An optimal velocity planning scheme for vehicle energy efficiency through probabilistic prediction of traffic signal timing. *Accepted for Publication in IEEE Transactions on Intelligent Transportation Systems*.
- [29] H. Xia, K. Boriboonsomsin, F. Schweizer, A. Winckler, K. Zhou, and W.B. Zhang. Field operational testing of ECO-Approach technology at a fixed-time signalized intersection. In *Proceedings of the 2012 15th International IEEE Conference on Intelligent Transportation Systems*, Anchorage, Alaska, September 2012.
- [30] A. Lawitzky, D. Wolherr, and M. Buss. Energy optimal control to approach traffic lights. In *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo Big Sight, Japan, November 2013.
- [31] E. Horvitz, J. Apacible, R. Sarin, and L. Liao. Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service. *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, July 2005.
- [32] J.L. Ygnace, C. Drane, Y.B. Yim, and R. de Lacvivier. Travel time estimation on the san francisco bay area network using cellular phones as probes. Technical report, California PATH Program, Institute of Transportation Studies, University of California, Berkeley, 2004. California PATH Working Paper UCB-ITS-PWP-2000-18.
- [33] Arvind Thiagarajan, Lenin Ravindranath, Katrina LaCurts, Sivan Toledo Jakob Eriksson, Samuel Madden, and Hari Balakrishnan. Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, Berkeley, California, November 2009.
- [34] A. Kovacheva, R. Frank, and T. Engel. Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *2013 19th IEEE Workshop on Local and Metropolitan Area Networks*, Brussels, Belgium, April 2013.
- [35] WAZE. Waze: Outsmarting traffic, together. <http://www.waze.com>.

- [36] S. Fayazi, A. Vahidi, G. Mahler, and A. Winckler. Traffic signal phase and timing estimation from low-frequency transit bus data. *Accepted by IEEE Transactions on Intelligent Transportation Systems*.
- [37] Wireshark. Wireshark. <http://www.wireshark.com>.
- [38] P.-S. Lin, A. Fabregas, H. Chen, and S. Rai. Impact of detection and communication degradations on traffic signal operations. In *ITE 2010 Annual Meeting and Exhibit*, Vancouver, Canada, August 2010. <http://www.ite.org/annualmeeting/compendium10/pdf/AB10H4002.pdf>.
- [39] G. Mahler, A. Winckler, M. Filusch, and A. Vahidi. Fuel efficiency improvements of connected vehicles on arterial roads. *Submitted to IEEE Transactions on Vehicular Technologies*.
- [40] Siemens: SEPAC Training Guide. Technical report, Siemens, 2005.
- [41] USDOT. Investment opportunities for managing transportation performance through technology, January 2009. [http://www.its.dot.gov/press/2009/transportation\\_tech.htm](http://www.its.dot.gov/press/2009/transportation_tech.htm).
- [42] P. Leviakangas and J. Lahesmaa. Profitability evaluation of intelligent transportation system investments. *Journal of Transportation Engineering*, 128:276–286, May/June 2002.
- [43] M. Maile, F. Ahmed-Zaid, S. Bai, L. Caminiti, P. Mudalige, M. Peredo, and Z. Popovic. Objective testing of a cooperative intersection collision avoidance system for traffic signal and stop sign violation. In *2011 Intelligent Vehicles Symposium*, Baden-Baden, Germany, June 2011.
- [44] M.R. Hafner, D. Cunningham, L. Caminiti, and D. Del Vecchio. Cooperative collision avoidance at traffic intersections: Algorithms and experiments. *IEEE Transaction on Intelligent Transportation Systems*, 14:3:1162–1175, September 2013.
- [45] B. Schweiger, R. Glas, C. Raubitschek, and J. Schlichter. Traffic signal information in a real residential area. In *Proceedings of the FISITA 2012 World Automotive Congress, Volume 12*, Beijing, China, November 2012.
- [46] P. Galaske. Entwicklung einer methode zur behandlung der unsicherheit von ampelphaseninformationen fur vorausschauende fahrassistentensysteme. Master’s thesis, Technische Universitat Darmstadt, 2012.

- [47] NextBus. <http://www.nextbus.com>.
- [48] Mobileye. Mobileye development and evaluation platforms. <http://www.mobileye.com/technology/development-evaluation-platforms/>.
- [49] J. Stevanovic A. Stevanovic and C. Kergaye. Impact of signal phasing information accuracy on green light optimized speed advisory systems. In *Transportation Research Board Annual Meeting 2013*, Washington, D.C., January 2013.
- [50] INTERSAFE. Web. [http://www.prevent-ip.org/en/prevent\\_subprojects/intersection\\_safety/intersafe/](http://www.prevent-ip.org/en/prevent_subprojects/intersection_safety/intersafe/).
- [51] M. Maile, F. Ahmed-Zaid, L. Caminiti, J. Lundberg, P. Mudalige, and C. Pall. Cooperative intersection collision avoidance system limited to stop sign and traffic signal violations. Technical Report DOT HS 811 048, NHTSA, Federal Highway Administration, October 2008.
- [52] R. Sengupta, S. Rezaei, S. Shladover, D. Cody, S. Dickey, and H. Krishnan. Cooperative collision warning systems: Concept definition and experimental implementation. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, 11(3):143–155, 2007.
- [53] C.-Y Chan and B. Bougler. Evaluation of cooperative roadside and vehicle-based data collection for assessing intersection conflicts. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 165–170, 2005.
- [54] A. Vahidi and A. Eskandarian. Research advances in intelligent collision avoidance and adaptive cruise control. *IEEE Transactions on Intelligent Transportation Systems*, 4(3):143–153, 2003.

# Appendices

# Appendix A

## Appendices

### A.1 Supplemental Experimental Setup Information

The process of analyzing fuel economy for a vehicle with and without traffic signal recommendations involves both in-vehicle software as well as data post-processing.

In all cases the vehicle is the same. The BMW 535i has been equipped with both supplemental and prototype development systems which are necessary aspects to the testing process. The 535i has x86 based boxes in the trunk, running screens in both the dash and the Multi-Media Interface (MMI). These screens are essentially computer LCD monitors. The dash and Heads-Up Display (HUD) are running a development version of FLORIS, a Flash-based program which integrates with the Controller-Area-Network (CAN) to replicate via software and LCD display, the functionality of typical gauge and trouble-light based dash displays. Typical CAN bus messages could include vehicle speed, engine RPM, oil pressure, coolant temperature and many other things. In addition, we are able to send messages to the FLORIS



display regarding color spectrums we wish to display on the speedometer. In the examples detailed in this manuscript we have used this functionality for displaying speed recommendations for passing through traffic signals.

Traffic signal data is coming from three sources. The first source is from Traffic Management Centers. The second source is from reverse-engineered probe-vehicle data, including public transportation buses in San Francisco. The third source is from the development vehicle's camera system.

Data coming from TMCs is sent to a third party to integrate so-called static and dynamic data. Static data is traffic signal GPS position, intersection geometry, and similar information which describes the intersection itself, and not the status of the intersection. The dynamic data is the presence or absence of vehicle calls, pedestrian calls, the active phases, the inactive phases, and the presence or absence of overriding public transportation or emergency vehicle transponders. Dynamic information from signal controllers updates once per second to the TMC in the cities in which we tested. The TMC has a direct connection with the third party, and the third party filters and forwards status-based updates to a BMW server. User-Datagram Protocol (UDP) messages are sent from the third party to BMW when something about the status changes; for example, the arrival of a pedestrian, the switching from active to inactive of a phase, or similar things. BMW maintains an SQL database for each city with signal status information for each phase of each signal about which information is known.

Data coming from reverse engineered probe data starts with collecting and recording public bus route data for the city of San Francisco. Data is provided

by NextBus, over an HTML interface. Positional data is combined with speed data to estimate when the bus passed through, or was stopped at, the intersection. This process is repeated for each bus over a significant time period to determine base timing data for pre-timed intersections (e.g. cycle time, nominal split, etc.). Then, to determine what phases are currently active at an intersection, analysis is done on recent bus passes through the intersection. Several passes are compared to determine which passes estimate traffic signal timing with the least difference. Those passes which estimate the closest timing to each other are used to determine where the clock signal is with respect to the nominal timing plan, and thus which phases are active. This predictive information regarding estimated phase updates is then sent, over a similar UDP interface, to BMW and is stored in an SQL database.

Data coming from camera systems has the opportunity to be similar to, but more accurate than, the bus data. The in-vehicle development camera system recognizes traffic signals using image-processing algorithms. This information is then compared to vehicle position and heading and the currently perceived phase status information recorded. The information is available over the CAN bus and is then sent to BMW over a 4G/LTE connection. However, because the camera perceives the actual signal status, errors in active phase information are different than the second data source. Like the other data sources, this is stored in an SQL database by BMW.

With data collected from the three data sources, the problem remains of how to get data to the vehicle. The vehicle maintains a 4G/LTE connection to the internet. The SQL server with signal status information may be accessed via this network connection. When the test vehicle turns on, it asks the server what the

current version of the known traffic signal list is. This is the previously mentioned static information, and is primarily a list of intersections to which the system has a connection, and GPS and geometry information about those intersections. The vehicle compares the server version information to the in-vehicle version information, and updates if necessary.

As the vehicle drives through an area where GPS positions start to match up with traffic signal GPS information in the maintained list of intersections, a selection algorithm starts running. This is the multi-step signal selection algorithm described in detail in Chapter 4. The upcoming traffic signal is selected, and a message sent to the server subscribing to that phase of that signal. The server responds with the appropriate status information, and any subsequent status changes. As the vehicle passes through the intersection, an unsubscribe message is sent to the server.

In the vehicle, this signal status information is used to create the velocity recommendations shown to the driver. As described in Chapters 2 and 4, the phase change information is used to calculate upper and lower bounds for passing through green, or being blocked by red. In addition, the camera information is used to determine whether obstacles will be impediments to the vehicle's safe following of those recommendations, and if so, only a phase countdown and not a velocity recommendation can be made.

A total of 14 drivers were evaluated using the in-vehicle display. Drivers were predominantly interns at the BMW Technology Office USA in Mountain View, California. As the people most closely related to the project, neither Andreas Winckler nor myself were test drivers. Drivers were asked to drive from the Tech Office to

the test area in San Jose, as an opportunity to familiarize themselves with the vehicle. As described in Chapter 4, drivers drove first with the system off and then with the system on. This ordering was done for two reasons. The first was that we did not want drivers to achieve the benefits of the system after it was turned off simply because they have muscle memory and knowledge of the signal timing from repeated driving of the test loop. Secondly, we did not want to tell drivers what we were testing, because we wanted them to drive naturally in the naive test session. Of the drivers, only 1 was female. Driver ages ranged from 18 to almost 40.

While driving, we record the full CAN bus data stream. We also record specific messages pertaining to GPS position, vehicle speed, fuel consumed, and several other things. We record separate files for the naive and the informed test drives. These files are analyzed in post-processing. We use scripts written in Matlab to visualize how many laps were driven, and to calculate fuel economy. The fuel economies calculated here are those reported in preceding chapters.