Final Manuscript, Accepted April 2014
To Appear in IEEE Transactions on Intelligent Transportation Systems



# Traffic Signal Phase and Timing Estimation from Low-Frequency Transit Bus Data

S. Alireza Fayazi  Ardalan Vahidi  Grant Mahler  Andreas Winckler

***Abstract*—The objective of this paper is to demonstrate the feasibility of estimating traffic signal phase and timing from statistical patterns in low-frequency vehicular probe data. We use a public feed of bus location and velocity data in the city of San Francisco as an example data source. We show it is possible to estimate, fairly accurately, cycle times and duration of reds for fixed-time traffic lights traversed by buses using a few days worth of aggregated bus data. Furthermore, we also estimate the start of greens in real-time by monitoring movement of buses across intersections. The results are encouraging, given that each bus sends an update only sporadically (≈ every 200 meters) and that bus passages are infrequent (every 5-10 minutes). When made available on an open server, such information about traffic signals' phase and timing can be valuable in enabling new fuel efficiency and safety functionalities in connected vehicles: Velocity advisory systems can use the estimated timing plan to calculate velocity trajectories that reduce idling time at red signals and therefore improve fuel efficiency and lower emissions. Advanced engine management strategies can shut down the engine in anticipation of a long idling interval at red. Intersection collision avoidance and active safety systems could also benefit from the prediction.**

## I. Introduction

Traffic signals have been an indispensable element of our transportation networks since their inception and are not likely to change form or function in the foreseeable future [1]. While traffic signals ensure safety of conflicting movements at intersections, they also cause much delay, wasted fuel, and tailpipe emissions. Frequent stops and goes induced by a series of traffic lights often frustrates drivers. In arterial driving, the complex and unknown switching pattern of traffic signals, makes accurate travel time estimation or optimal routing often impossible even with modern traffic-aware in-vehicle navigation systems. Much of these difficulties arise due to the lack of information about the current and future state of traffic signals. In an ideal situation where the state of a light's timing and phasing is known, the speed could be adjusted for a timely arrival at green [2]. One can expect considerable fuel savings in city driving with such predictive cruise control algorithms as shown in [2] and [3]. When idling at red becomes unavoidable, knowledge of remaining red time can determine if an engine shut-down is worthwhile. A collision warning system can benefit from the light timing information and warn against potential signal violations [4]. Future navigation system that have access to the timing plan of traffic lights, can find arterial routes with less idling delay [5] and can also provide more accurate estimates of trip time.

The main technical challenge to deploying such in-vehicle functionalities is in reliable estimation and prediction of Signal Phase And Timing (SPAT): Uncertainties arising from clock drift of fixed-time signals, various timing plan of actuated traffic signals, and traffic queues render this a challenging and open-ended problem. Direct access to signal timing plans and real-time state of the light is prohibitively difficult due to hundreds of local and federal entities that manage the more than 330,000 traffic lights across the United States alone [6]. Even when such access is granted, much effort and time must be spent in structuring information from various municipalities in standard and uniform formats. The more recent emphasis on Dedicated Short Range Communication (DSRC) technology for communicating the state of traffic signals to nearby vehicles has safety benefits, but requires heavy infrastructure investments and even then is limited by its short communication range.

To overcome some of these difficulties, in this paper we propose an alternative approach, that relies on vehicle probe data streams, for estimating a signal's phase and timing. In recent years several research groups have shown that mobile phone or vehicle probe data can be effectively utilized for estimation of traffic flow [7], [8], [9]. Today many traffic information providers, such as Google, INRIX, and Waze use data from vehicle and cellular phone probes, as well as other means, to estimate the severity of traffic on highways nearly in real-time. However such algorithms perform relatively poorly in arterial networks because traffic signals induce complex queue and stop and go dynamics. Some more recent work has focused on estimating queue lengths [10] and on determining location of traffic signals and stop signs [9] through use of vehicle probe data. What seems to be missing from the literature is a systematic attempt to derive SPAT information from available vehicle data streams. The only related work that the authors are aware of is [11] in which a simulation study is performed to show feasibility of determining SPAT using probe data. What limits the results in [11] is its assumption on frequency of data update (≈1Hz) and expectation that the penetration level is high.

Unfortunately, currently one cannot expect high update rates from public fleets that broadcast their information, nor is there a proliferation of vehicle probes. Most existing ones only provide event-based updates, for example at a time of a crash or air-bag deployment. Interesting data sources such as San Francisco taxi cab data available through the cab-spotting program [12] have update rates of only once per minute. More

S. A. Fayazi (corresponding author, sfayazi@clemson.edu), A. Vahidi (avahidi@clemson.edu), and Grant Mahler (gmahler@clemson.edu) are with the Department of Mechanical Engineering, Clemson University, Clemson, SC 29634-0921, U.S.A. Andreas Winckler (andreas.winckler@bmw.de) is with the BMW Group, Munich, Germany.

frequent updates are available through NextBus, a service that provides a real-time XML feed of GPS time stamp, position, velocity, and several other attributes of transit buses of a few cities in North America [13]. Some instances of this feed, such as San Francisco MUNI stream, have update rates on the order of twice per minute. And one can be certain that intersections along a bus route get traversed by a bus every few minutes during the day. An open question that we try to address in this paper is how much, statistical patterns in such low-frequency data can reveal about the state and parameters of traffic lights. This determines what the minimum achievable is; as higher frequency probe data becomes available in the future, more accurate estimates of parameters of traffic signals can be obtained.

After a short description of the NextBus data stream in Section II, we explain reconstruction of the approximate trajectory of a bus between each two update points in Sections III and IV. Section V presents our methodology and results for estimation of red time and cycle time of a traffic signal based on available and reconstructed bus data. We also discuss the potential for extracting other attributes such as an estimate of the signal clock time (start of greens) in Section VI, changes in a signal's offset and schedule in Section VII, and probability of green in Section VIII. We will compare our estimates versus the ground truth measurements at an intersection in the city of San Francisco in Section IX. Section X provides concluding remarks.

## II. Description of the Data Feed

The results in this paper are based on data from bus movements in the city of San Francisco. The bus data feed is provided by NextBus [13] for a number of cities in North America in eXtensible Markup Language (XML). The attributes of interest are position and velocity of each bus along with their time stamp and the bus identification number. Also the bus route data and location of bus stops are extracted from the same data stream. A map of bus (and light rail) routes in San Francisco in Figure 1 is constructed by aggregating GPS updates from all buses within a twenty-four hour period. The

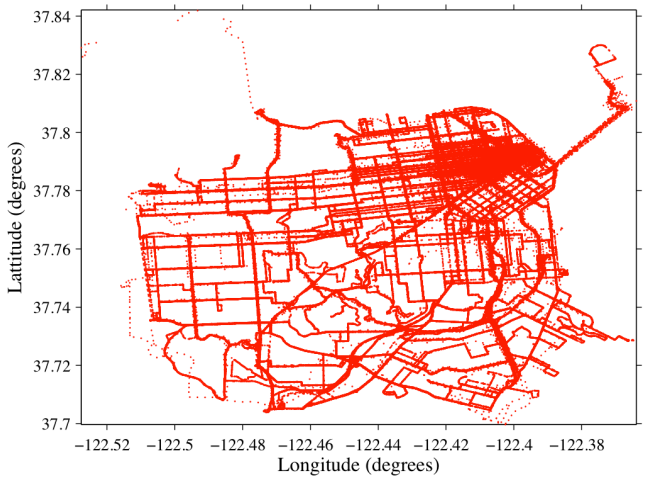


Fig. 1. *Aggregated plot of all bus (MUNI) updates for a period of 24 hours in the city of San Francisco.*

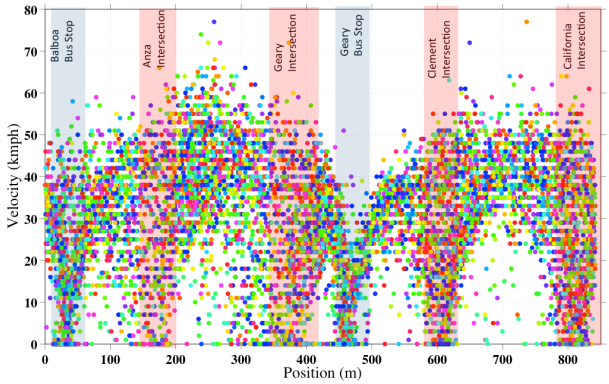


Fig. 2. *Scatter plots of San Francisco Route 28 bus updates over one month (September 2012). A total of 2478 bus passes are shown.*

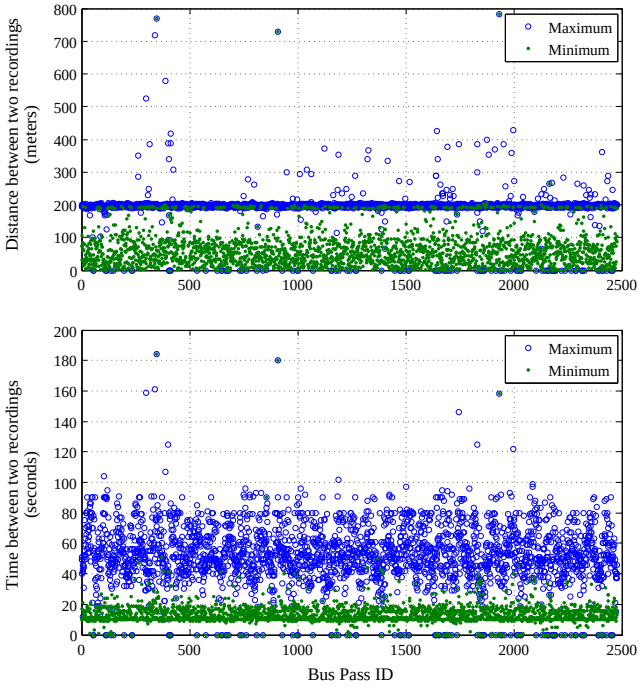


Fig. 3. *Maximum and minimum distance and time between two updates of San Francisco Route 28 buses over one month (September 2012) along the short portion of Park Presidio Blvd depicted in Figure 2.*

focus on this paper is only on a few bus routes to show the feasibility of the proposed ideas.

Figure 2 shows example data from a portion of bus route 28 along Park Presidio Boulevard in the city of San Francisco. This is an aggregation of 2478 bus passes over an entire month. While each bus sends only four or five updates along the shown stretch of the route, the aggregated data is very revealing and correctly depicts the location of intersections and bus stops. Figure 3 shows the maximum and minimum distance and time between two updates of each bus pass and for every one of the 2478 bus passes. According to this data, the updates do not seem to be at regular time or distance intervals. Time updates are anywhere between every 10 seconds up to every 80 seconds or sometimes more. However there is a strong concentration of data at 200 meters distance intervals which indicates that most updates happen every 200 meters. From these update rates it seems that slower

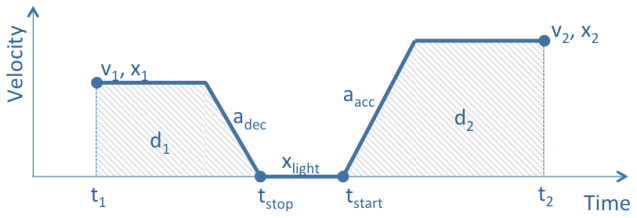


Fig. 4. *Reconstruction of a bus trajectory that stops at an intersection.*

buses update at shorter distance intervals based on a time threshold.

## III. Reconstructing Bus Kinematics from Sparse Data

We would like to estimate if a bus was stopped at an intersection, how long it was stopped, and at what time it left the intersection. We hope by aggregating this information for many buses we can estimate the duration of a red phase, the cycle length, the start of a green phase, and perhaps more. But because the update points for each bus are sporadic, we need to approximate a bus trajectory between each two update points. The following steps are followed:

- Step 1: For a given intersection, we first select bus passes that have update points within a given interval before and after that intersection. For example for the Clement Intersection shown in Figure 2, after observing the trend in the aggregated data, we select bus passes that updated in both [480m, 590m] and [620m, 780m] position intervals. Furthermore we filtered out also passes with low velocity (less than 5 km/h for results in this paper), to ensure that the influence of heavy traffic is minimized on signal timing estimation.
- Step 2: To determine if a bus stopped at an intersection, we propose to approximate the intersection delay, $t_d$, by subtracting projected travel time from actual travel time as follows:

$$t_d = (t_2 - t_1) - \frac{x_2 - x_1}{(v_1 + v_2)/2} \tag{1}$$

where $x_1$, $v_1$, and $t_1$ denote the position, velocity, and time stamp of the last update of a bus before an intersection of interest, and $x_2$, $v_2$, and $t_2$ are the position, velocity, and time stamp of the first update of that bus after the intersection. Therefore $t_2 - t_1$ is the actual travel time and $\frac{x_2-x_1}{(v_1+v_2)/2}$ is the estimated travel time if the velocity of the bus had changed linearly between $v_1$ and $v_2$.
If $t_d \leqslant 0$, we postulate that the bus had no delay and that it passed the intersection during a green interval. Otherwise, we may attribute the delay to a stop at red, which will be further confirmed in the next step.
- Step 3: When $t_d > 0$, we check the consistency of the trajectory shown in Figure 4 with data. In other words, we approximate that the bus moves with a constant velocity $v_1$, then comes to a stop at the intersection at a constant deceleration $a_{dec}$, and then at start of a green it accelerates with constant acceleration $a_{acc}$ to a constant velocity $v_2$. If the location of the light $x_{light}$, is known, then $d_1 = x_{light} - x_1$ and $d_2 = x_2 - x_{light}$ are areas under the the time-velocity curve. Using the trapezoidal geometry of the curves, we can then estimate the time a bus comes to a stop $t_{stop}$ and the time the bus leaves the intersection $t_{start}$ as follows:

$$t_{stop} = t_1 + \frac{d_1}{v_1} + \frac{v_1}{2a_{dec}} \tag{2}$$

$$t_{start} = t_2 - \frac{d_2}{v_2} - \frac{v_2}{2a_{acc}} \tag{3}$$

Obviously if $t_{stop} > t_{start}$, the postulated trajectory is invalid and the associated bus pass will be discarded. When $t_{stop} \leqslant t_{start}$, we accept the trajectory as valid and estimate that the bus came to a full stop at a red light. The duration of red "observed" by a particular bus is then estimated as:

$$t_{red} = t_{start} - t_{stop} + \frac{v_1}{a_{dec}} \tag{4}$$

where $\frac{v_1}{a_{dec}}$ is the time it takes a bus to come to a full stop after the driver detects the signal is red. Aggregating $t_{red}$ for a sufficiently large number of bus passes will later lead to an estimate of total red duration of a phase.
In the above calculations we assumed that acceleration and deceleration of buses were known and constants. We show next how probe data is used to approximate the average acceleration and deceleration of the bus fleet. We also demonstrate that $t_{red}$ is not highly sensitive to reasonable variations in the value of acceleration.

## IV. Crowd-Sourcing Acceleration and Deceleration of Buses

Because of data sparsity, it is not possible to estimate the acceleration or deceleration of an individual bus. However velocity-position data from many buses shows a trend in

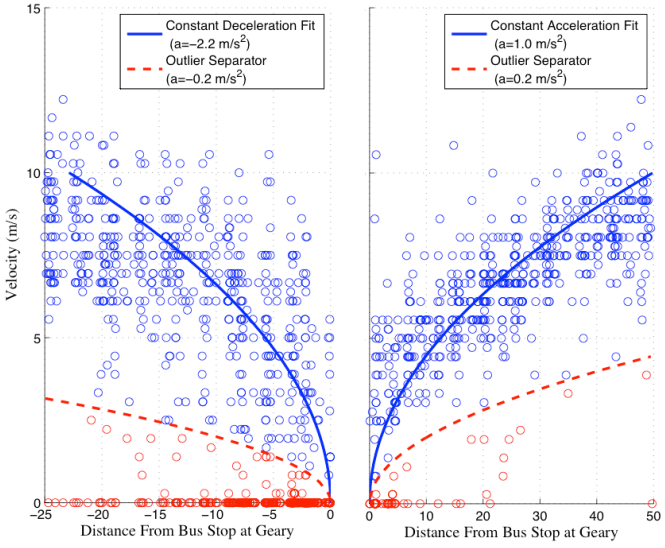


Fig. 5. *Estimation of average deceleration and acceleration of buses during stop and start using probe data.*

start/stop trajectory as seen in Figure 2. For instance, at the Geary bus stop where a majority of buses come to a full stop, one can observe a clear slow-down and speed-up trend which can be used to estimate an average value for a bus deceleration and acceleration, later shown in Figure 5. To simplify the future steps of this work, we assume that deceleration to a stop and acceleration from a stop for a bus are constants and not functions of velocity. Hence the velocity while accelerating from a stop at a signal can be related to the distance traveled as follows:

$$v^2(x) = 2\bar{a}_{acc}(x - x_{signal}) \tag{5}$$

where $\bar{a}_{acc}$ is the average acceleration which is to be estimated from data. A similar equation can be written for a deceleration interval. By defining $y = x - x_{signal}$, $\psi = v^2(x)$, and $\theta = \frac{1}{2\bar{a}_{acc}}$ Equation (5) can be reorganized in the following linear parameterized form:

$$y = \theta\psi \tag{6}$$

Several data points can be stacked in a least-square approach to estimate the parameter $\theta$ and therefore $\bar{a}_{acc}$. As seen in Figure 5 there are several outlier data points that will skew the estimation result. So in the least square estimation, we have ignored the data points (in red) below a certain acceleration/deceleration profile (shown by dashed curves) to reduce the influence of outliers. Figure 5 shows the resulting curve fit for both deceleration and acceleration. The estimated deceleration is 2.2 m/s$^2$ and the estimated acceleration is 1.0 m/s$^2$. These values are consistent with bus acceleration measurements reported in [14], [15][1].

## V. Estimating a Signal's Baseline Timing

The goal in this section is to determine if the baseline timing for lights can be obtained by *offline* aggregation and averaging of crowd-sourced bus data. In particular, we are interested in determining the duration of reds/greens of a phase and the cycle time of a traffic signal. Later we will investigate if a signal's clock time and schedule changes can be calculated. But we note that mere knowledge of baseline schedule, obtained offline and using only historical data, has statistical value even when a signal's clock-time is unknown. See for example [16] in which the baseline schedule of a light is used to predict the chance of a future green for an eco-driving application.

While we have results from several intersections in different locations in San Francisco, in the rest of this paper we focus on results for a segment of Van Ness street, between Lombard and Bush intersections. This is a sometimes congested street and therefore suited to test our proposed algorithms under (relatively heavy) city traffic conditions. Additionally, we have access to the actual signal timing cards of intersections of Van Ness and therefore can verify the validity of our estimates. Most intersections on this segment of Van Ness are fixed time intersections with the same cycle time and red duration throughout all days of the week. For most of these traffic signals, only offset times change during rush hour schedule, that could be estimated as we show later in this paper. We aggregate one month worth of data (September 2012) from two bus routes, route 47 and route 49, in the southbound direction totaling 4289 bus passes. This data is used to estimate signals' cycle time and the timing of the phases controlling southbound traffic on Van Ness, as explained next.

### A. Estimating Duration of a Red Phase

For each bus pass we follow the procedure explained in Section III and for those that had stopped at a red, the observed red time is calculated via Equation (4). Aggregating this data provides an estimate of the duration of red for the corresponding phase. For example for the southbound phase on Van Ness street at Lombard intersection, there remained 347 bus passes after applying the filters described in Section III to the 4289 total passes. Figure 6 presents the observed red for these 347 passes in two forms: The histogram of observed reds in the first subplot has a maximum of 68 seconds which is an upper bound estimate to duration of red phase. The second subplot shows the observed reds at different hours of a day for an entire month. During early morning hours (midnight-6am) and late night hours (7pm-11pm) where the queue lengths are expected to be shorter, we observe a maximum observed red of 60 seconds. This corresponds well to the actual timing of this intersection: According to the city timing cards, this intersection has a 90 second cycle time split to 60 seconds of red, 3.5 seconds of yellow, and 26.5 seconds of green for the southbound phase. Note also that many bus drivers may treat a yellow as red increasing their observed red time to a maximum of 63.5 seconds. We repeated this process for a few other intersections on Van Ness and the results are summarized in Table I. In most cases the red estimates are very close to the actual red. This is while, unlike Lombard Intersection, many of these intersections had a short red interval and a green-wave design that allowed most buses to pass through their green period; thus offering a smaller number of usable data points[2].

### B. Estimating Cycle Times

For fixed-time signals with phases that repeat cyclically, the time between start of greens of a phase must be an integer multiple of the cycle time[3]. An approximation for a start of green can be obtained using Equation (3), i.e. the clock time that a bus starts accelerating from a stop at red. The difference between two consecutive approximations of start of greens,

[1]The sensitivity of $t_{red}$ estimate in Equation (4) to variations in acceleration (also similarly deceleration) can be found to be:

$$\delta t_{red} = -\frac{v_2}{2}\frac{\delta a_{acc}}{a_{acc}^2}$$

and because $v_2$ is at most around 20 m/s for a city bus and $a_{acc}$ and $a_{dec}$ are greater than 1 m/s$^2$, even a 20% error in approximation of $a_{acc}$ ($\delta a_{acc}/a_{acc} = \pm 0.2$) results in a maximum error of 2 seconds for $t_{red}$. The error is much smaller in most places where $v_2$ is much less than 20 m/s.

[2]A part of the larger error at Broadway intersection may be due to the steeper slope of Van Ness street at Broadway intersection which is not taken into account in crowdsourcing acceleration and deceleration of the buses.

[3]Note that due to a signal's clock drift this may not be true for start of greens that are far apart.

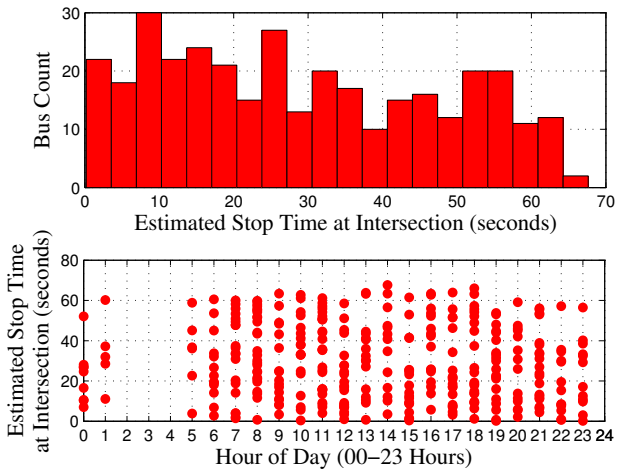


Fig. 6. *Stop time at red by each probe vehicle a) histogram b) stop time at different times of day. Southbound through phase on Van Ness Street at Lombard Intersection.*

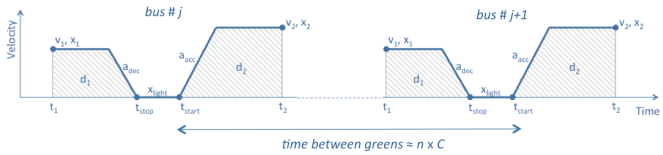


Fig. 7. *Time between consecutive start of greens must be an integer multiple of cycle time for a fixed-cycle traffic signal.*

based on bus movements, then must be an "almost" integer multiple of the cycle time, as shown schematically in Figure 7. Let's denote the time between approximated start of greens as $b_g$, therefore,

$$b_g(j) = t_{start}(j+1) - t_{start}(j) \tag{7}$$

For a given cycle time $C$, we can then calculate the remainder of division of $b_g$ and $C$ as follows:

$$\text{mod}_C(b_g) = b_g - \text{round}(b_g/C)C \tag{8}$$

where the function round(.) rounds its argument to the nearest integer and the function $\text{mod}_C(.)$ is a modified definition of remainder of division by $C$ that allows negative values. For example $\text{mod}_{10}(12) = 2$ and $\text{mod}_{10}(8) = -2$.

We expect $\text{mod}_C(b_g)$ to be close to zero on average, if the cycle time is fixed at $C$ and signal clock drift between two qualifying bus passes is small. Therefore we propose to approximate $C$ by solving the following optimization problem:

$$\bar{C} = \arg\min_C \sum_{j=1}^{n} \left( \frac{\text{mod}_C(b_g(j))}{C/2} \right)^2 \tag{9}$$

where it is assumed there are $n+1$ qualifying bus passes during the interval of interest and therefore $n$ calculations of $b_g$. Observing that $-\frac{C}{2} < \text{mod}_C(.) \leqslant \frac{C}{2}$, we normalize the remainders by $C/2$ to ensure all values of $C$ generate equivalent costs.

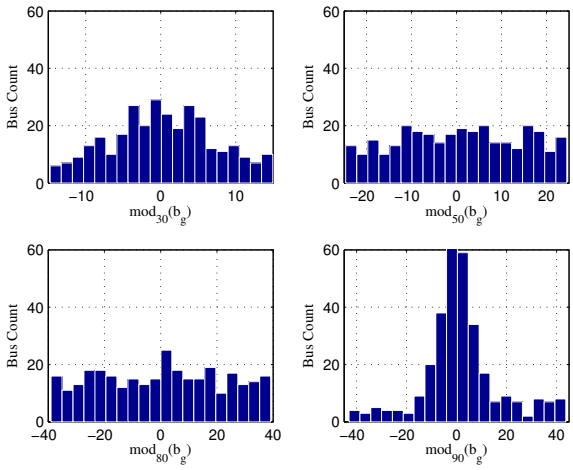


Fig. 8. *Deviation of approximated time between start of greens from multiples of example cycle times. At the actual cycle time of $C = 90$ seconds, a clear peak can be observed.*

Because a signal cycle time is normally an integer in practice and has a limited range, one can conveniently solve the above optimization problem by trying every feasible $C$. We tried integer values between 1 and 120 seconds when determining cycle time of signals on Van Ness. To reduce the influence of signal clock drift we limit the choice of $b_g$ to those within a few hours, e.g. 5 hours for results in this paper. Using one month worth of data, the estimated cycle time for Lombard intersection was 90 seconds, perfectly matching its actual value. This is visually illustrated in Figure 8 with histograms of $\text{mod}_C(b_g)$ for Lombard Intersection for four different values of $C$. As it can be seen, for $C = 90$ seconds, the histogram peaks strongly around zero despite various sources of uncertainty, i.e. unknown queue lengths and traffic conditions and approximations made in reconstructing bus trajectories. In the fourth subplot, we also observe small bumps near the tail ends; later in Section VII, we explain that these bumps are direct results of change in signal offset times during rush hour schedules.

TABLE I
RED AND CYCLE TIME ESTIMATES FOR A FEW SOUTHBOUND PHASES THROUGH VAN NESS STREET, CALCULATED USING DATA FROM BUS ROUTES 47 AND 49 GATHERED FOR SEPTEMBER 2012.

| Intersection | Actual Red (seconds) | Estimated Red (seconds) | Actual Cycle (seconds) | Estimated Cycle (seconds) | Qualifying Passes (count) |
|---|---|---|---|---|---|
| Lombard | 60 | 60 | 90 | 90 | 347 |
| Filbert | 31.5 | 30 | 90 | 90 | 170 |
| Green | 31.5 | 35 | 90 | 90 | 86 |
| Broadway | 36 | 42 | 90 | 90 | 133 |
| Washington | 31.5 | 32 | 90 | 45 | 94 |
| Bush | 31.5/38.5 | 38 | 75/90 | NA | 41 |

Table I summarizes cycle estimates for a number of other intersections along Van Ness. For most, the estimated and actual cycle times are identical. For Washington Intersection, our proposed algorithm estimates the cycle time at exactly half of its actual value. This is partly due to lack of enough qualifying bus passes for this intersection. There were only 94 bus passes that qualified the filters for Washington as compared to 347 passes for Lombard Intersection. Also we were not able to obtain meaningful results for Bush intersection which is an actuated intersection with two different cycle times. Bush

Intersection had also very few (41) qualifying bus passes, as it was mostly green to buses traveling southbound.

## VI. Estimating Start of Greens

For real-time in-vehicle applications, it is important to have an estimate of the start of future green (or red) phases. Estimating the start of a green is a challenging problem: even for fixed-time signals that have fixed cycles, periodic projection of start of greens can be inaccurate due to signal clock drift throughout a day. To address this problem, we propose to continuously estimate the start of a green phase based on the movement of buses that accelerate from a stop at an intersection. In other words, Equation (3) can be used to estimate the time $t_{start}$ that each bus left the intersection. A moving average of the most recent times, can then be used to estimate the start of a green. More specifically, because of $C$-periodicity of a fixed-time light within each schedule, we can map the latest estimates of start of green to a single reference interval $[-\frac{C}{2}, \frac{C}{2}]$ by applying the $\text{mod}_C$ operator, e.g. for the $i^{th}$ qualifying bus pass:

$$t_i = \text{mod}_C(t_{start}(i)) \tag{10}$$

We can then create an average estimate of the start of green in this reference interval. Note that, a simple "linear" average will, in general, produce an erroneous estimate due to the cycle periodicity. See for examples the schematic in Figure 9 where

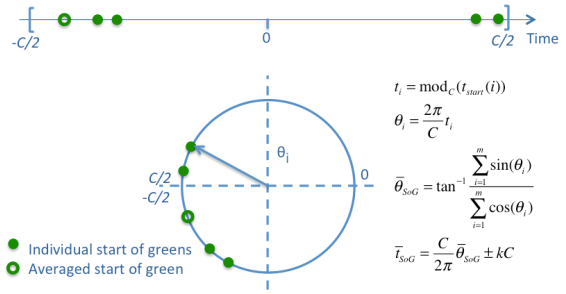


Fig. 9. *Schematic: Start of greens mapped to a reference C-periodic interval for calculating the average and standard deviation of start of greens.*

four estimates of green, mapped to the linear interval, and their true average are shown on a straight line. As seen in this example, the correct average does not fall between the individual greens. The periodicity can be better visualized if the time axis is wrapped onto a circle shown in Figure 9. Each start of green can then be represented by a vector with angle $\theta_i = \frac{2\pi}{C} t_i$ on the circle. The average angle, $\bar{\theta}_{SoG}$, is determined by the direction of the vector sum of all individual vectors:

$$\bar{\theta}_{SoG} = \tan^{-1} \frac{\sum_{i=1}^{m} \sin(\theta_i)}{\sum_{i=1}^{m} \cos(\theta_i)} \tag{11}$$

here $m$ represents the number of samples used to calculate the moving average. The average start of the green is obtained by mapping back, the average angle to the time axis:

$$\bar{t}_{SoG} = \frac{C}{2\pi} \bar{\theta}_{SoG} \pm kC \quad k \in \mathbb{Z} \tag{12}$$

The variance of this estimate is then obtained based on the minimum cyclic distance to the average, equivalently calculated by:

$$\sigma^2_{SoG} = \frac{1}{m} \sum_{i=1}^{m} (\text{mod}_C(t_i - \bar{t}_{SoG}))^2 \tag{13}$$

We will show later in Section IX that, in some instances, the accuracy of $\bar{t}_{SoG}$ can be enhanced, if we selectively choose samples that produce smaller variances. In other words with $n$ latest samples, we propose to calculate $\bar{t}_{SoG}$ and $\sigma_{SoG}$ for all possible combinations of $m < n$ samples and select the one with the minimum variance.

## VII. Estimating Changes in Signal Schedule

The traffic signals that we have considered on Van Ness street have 3 different schedules. While cycle times remain constant across multiple schedules for these intersections, each signal's offset with respect to other signals and also with respect to a reference clock switches as the schedule changes. For example at Lombard intersection and during weekdays, the start of the cycle is moved backward by 34 seconds at 6 AM and at 3 PM and moved forward at 10 AM and 7 PM. It is essential to estimate the change in offset and time of this change, if we are to solely rely on crowd-sourced data for predicting the start of a green. Here we report a couple of methods that were relatively successful in estimating time of change and amount of offset.

### A. Estimating Time of a Schedule Change

We propose to detect a change in signal offset/schedule by keeping track of start of greens and detecting when a start of green shifts off significantly from its periodic prediction.
A smaller value of variance calculated in Eq. (13) indicates that the corresponding $m$ estimates of start of green are consistent with each other and multiple of $C$ seconds apart. Right after a schedule change when the start of greens are shifted by the offset times, the variance is expected to temporarily increase, until it is corrected by newer estimates of start of greens. Jumps in the value of variance can then be indications of a change in signal schedule/offset times.

To test this hypothesis, we combined three months worth of data and calculated the variance of the moving average as a function of time of day[4]. Figure 10 shows the results for the intersection with Lombard for every day of the week. One can see clear jumps in the value of variance at 6 and 10 AM, and at 3 and 7 pm on a weekday. These correspond to the times that the signal schedule changes. For some days of the week there is also a large spike at around 8 AM; these spikes do not correspond to a schedule change, but perhaps are results of heavier traffic at that time. The plots for weekends do not have major spikes, which is consistent with the single schedule that is in effect on weekends. We conclude that spikes that happen recurrently on all weekdays are considered to correspond to signal schedule change while non-recurrent spikes may be due to heavy traffic.

[4]A first attempt to only use a couple of weeks worth of data had many gaps due to sparsity in qualifying bus passes.

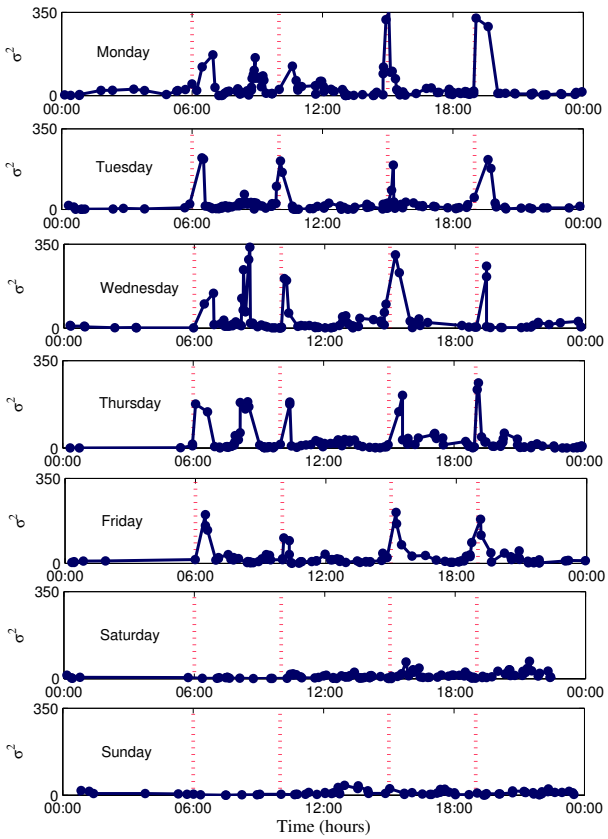


Fig. 10. *Variance of moving average estimate of start of green at different times and days of the week for Lombard intersection. The jump in variance corresponds, most often, to the change in signal schedule at 6 and 10 AM and 3 and 7 PM (shown by dashed vertical lines) on weekdays.*

### B. Estimating Signal Offset

In the histogram corresponding to $C = 90$ seconds in Figure 8, there were small bumps near the tail ends that were not explained in Section V-B. Using the method of Expectation Maximization (EM) [17] we fitted a Gaussian mixture model to the histogram in Figure 8 and the result is plotted in Figure 11. EM found three distinct Gaussian clusters with parameters shown in Table II. The major cluster is centered almost at zero, which was expected; and the two minor clusters are centered at almost $\pm 30$. These correspond closely to the 34 second shift in timing of the signal during a schedule change. We have further verified this hypothesis, by identifying time of days at which $\text{mod}_{90}(b_g)$ exceed $\pm 30$ seconds. In nearly all cases, this happens across multiple schedules, enforcing our hypothesis that the tail bumps are due to signal offset. In this case, the mean of this minor clusters can be used as an estimate to the amount of schedule offset.

TABLE II
PARAMETERS OF THE GAUSSIAN MIXTURE FIT TO HISTOGRAM OF FIGURE 8

| mean ($\mu$) | standard deviation ($\sigma$) | weight ($\pi$) |
|---|---|---|
| -30.78 | 7.32 | 0.07 |
| -0.24 | 7.02 | 0.79 |
| 29.79 | 9.32 | 0.14 |

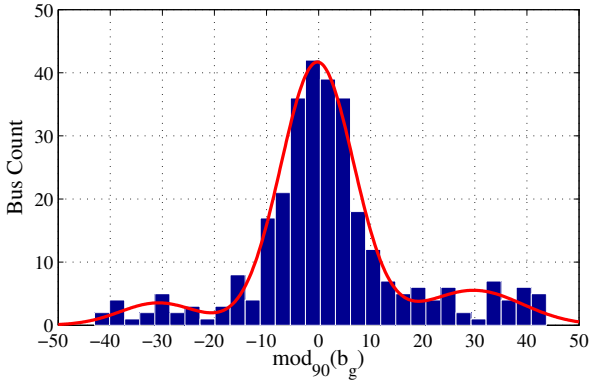


Fig. 11. *A Gaussian mixture model fitted to data of Figure 8 using the Expectation Maximization Algorithm. The peaks at tail ends correspond to the change to signal offset when schedule changes.*

## VIII. Direct Estimation of Green Intervals and Probability of Green

So far, all of our analysis has been based on movement of buses that had stopped at an intersection. We filtered out bus passes that had no intersection delay, e.g. those that cruised through a green. This approach discards a substantial amount of data, in particular for phases that either are often green or are timed in a green wave. But there is useful information that can be extracted from passes during a green: It is possible to interpolate a point in time that a phase was green based on the bus data before and after an intersection. Going back to Figure 4 and given the two update tuples $[t_1, x_1, v_1]$ and $[t_2, x_2, v_2]$ across one intersection, we propose the following steps:

- Step 1: Determine instances for which intersection delay calculated via Equation (1) is zero[5]. A zero value for $t_d$ indicates (with high likelihood) that the bus passed through a green and moreover, its acceleration between two update points remained constant.
- Step 2: Interpolate between update times $t_1$ and $t_2$ to determine the point in time at which the signal was green. For the constant acceleration case, we have:

$$x_{signal} = x_1 + v_1(t_g - t_1) + \frac{1}{2}a(t_g - t_1)^2 \qquad (14)$$

where $a = \frac{v_2 - v_1}{t_2 - t_1}$ is the constant acceleration between two update points. Here $t_g$ denotes a time at which the signal was green which is the feasible solution to the above quadratic equation:

$$t_g = t_1 + \frac{-v_1 + \sqrt{v_1^2 + 2a(x_{signal} - x_1)}}{a} \qquad (15)$$

- Step 3: Ideally we would like to aggregate all point calculations of $t_g$ to estimate intervals of green. For signals with fixed and known cycle time $C$, this can be done by mapping all values of $t_g$ onto a reference interval $[0, C]$.

[5]We used a small threshold and accepted values sufficiently close to zero.

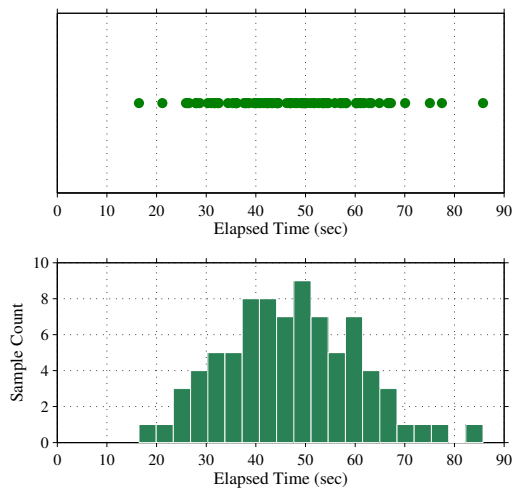


Fig. 12. *Green times mapped to one cycle interval. Southbound through phase on Van Ness Street at Lombard Intersection with cycle time of 90 seconds. Actual red was 60, actual green 26.5, and yellow 3.5 seconds.*

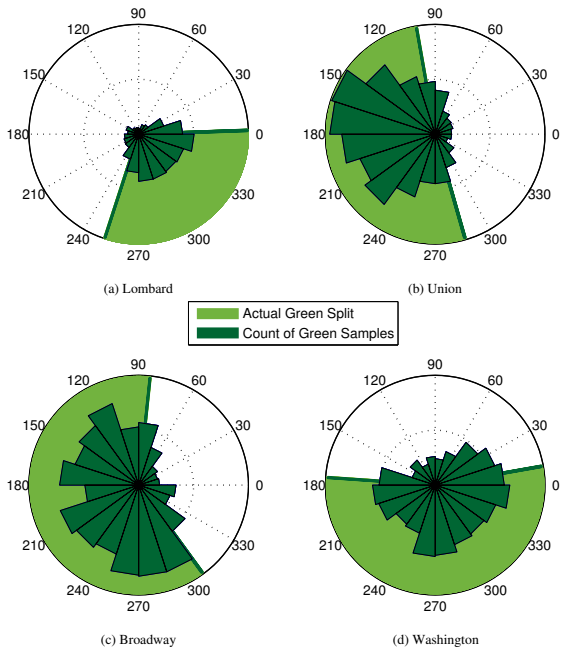


Fig. 13. *Crowd-sourced and actual green times mapped to one circular cycle interval in polar histograms. Southbound through phase on Van Ness Street at four different intersections.*

We carried out the above process for Lombard Intersection and the result is shown in the first subplot of Figure 12. When mapping all green times to a single interval, we have accounted for known changes in signal schedule. The second subplot is a histogram highlighting the concentration of points. In the ideal situation when a signal had no clock drift and repeated the same state at the exact same time every day, this mapping would result in an interval of green exactly matching signal's green time; i.e. 26.5 seconds for Lombard. But since the signal clock drifts, and also due to errors in reconstructing bus kinematics, the plotted green interval has a wider range than the actual green time. However there is much stronger concentration of mapped greens in the middle as shown by its histogram. This time period, and periods cyclically mapped forward, are where the probability of green is the highest. Even in the absence of any further crowd-sourced data, this probabilistic information is useful for many in vehicle applications (see [16] for instance).

Because of the cyclic periodicity, the data can be better visualized if mapped onto a polar histogram in which one revolution corresponds to one cycle time. Figure 13 shows such polar histogram plots for four different intersections along Van Ness. The height of each triangle represents the number of green samples within that triangle interval. Also shown by shaded areas on these plots are the actual green intervals, as observed and recorded in ground truth observations. It can be seen that the actual and crowd-sourced estimates of green interval match relatively well. The differences can be attributed to signal clock drift and also to errors in generating the crowd-sourced estimates.

## IX. Estimated Signal Clock Time versus the Ground Truth

To determine the accuracy of our estimates, in particular the start of greens, we arranged a session of on-site ground truth tests at the intersection of Lombard and Van Ness streets on June 6, 2013. Between the hours of 7 AM and 4 PM, we recorded the actual start of a green of the southbound phase on Van Ness almost every 15 minutes as the ground truth. This was done with the aid of a computer program that upon a key press would log the time as synchronized with the NIST time server [18]. The human observer's reaction time was determined to be less than 0.3 seconds which is sufficiently accurate for the purpose of this study.

Concurrently, the start of greens were estimated using the bus data feed and based on the procedure explained in Section VI. This was done in real-time via a crowd-sourcing backend server. The XML updates from routes of interest are continuously parsed and the data is written to a SQL data server. Another computational node constantly monitors the data to estimate start of greens and records it back on the SQL server. We could monitor the agreement between actual start of greens and crowd-sourced start-of-greens, in real-time, via a PHP web-interface.

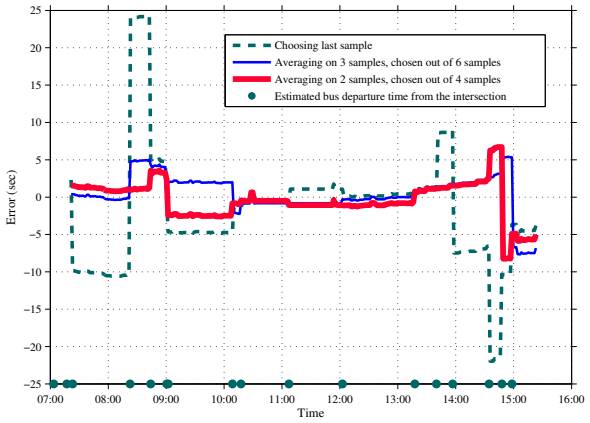


Fig. 14. *The error between crowd-sourced and actual start of greens for the Van Ness southbound phase at Lombard intersection as recorded on June 6, 2013. Green circles highlight times of qualifying bus passages.*

After each qualifying bus pass, new estimates for start of greens were generated using i) the last data point only, ii) minimum-variance average of 3 samples chosen out of last 6 data points, and iii) minimum-variance average of 2 samples chosen out of last 4 data points. Note that crowd-sourced estimate of greens are sparse in time due to the fact that the bus data that qualifies our filters is infrequent. Therefore in between two actual estimates, the start of greens are cyclically mapped using the estimated cycle time of the traffic light. Also the change in signal offset during schedule change is accounted for in this process. The estimated values for start of greens are then compared to the actual ground readings of the start of greens[6].

Figure 14 demonstrates the error between the crowd-sourced and actual start of greens. The jumps in error plots in Figure 14 correspond to the times when a new qualifying bus pass occurs. The drift in between is due to the actual drift of the signal clock and is not a by-product of crowd-sourcing. The root-mean-square and maximum error of each estimation approach are summarized in Table III. It can be observed that the minimum variance estimates are reasonably close to the actual timing with an RMS error of around 2.5 seconds. The estimate that was based on only last sample was more prone to error in this case.

TABLE III
ROOT-MEAN-SQUARE AND MAXIMUM ESTIMATION ERROR FOR START-OF-GREENS

| Estimation Method | RMS Error (Sec.) | Max. Error (Sec.) |
|---|---|---|
| Last data point | 8.0 | 24.3 |
| 3 out of 6 data points | 2.6 | 7.7 |
| 2 out of 4 data points | 2.5 | 8.2 |

## X. CONCLUSIONS

In this paper we demonstrated the feasibility of estimating timing of fixed time traffic lights by observing statistical patterns in sparse probe vehicle data feeds. In particular we showed, for example intersections in the city of San Francisco, the feasibility of estimating cycle time, red time, start of green, and signal schedule change. This was achieved without directly estimating the queue lengths and despite traffic influence. Extensive use of data filtering / pre-processing is elemental to the successes found at the given intersections. It should be noted that the influence of the heavy traffic conditions on the estimates is not investigated int this paper; nor did we consider actuated or adaptive signals. Our future work will focus on using of advanced statistical inference techniques, allowing us to make use of a larger portion of data to infer timing of the lights and perhaps also queue lengths formed behind each traffic light. As higher frequency probe data becomes available, we expect to obtain more accurate estimates of parameters of traffic signals, even those with actuated or adaptive controllers.

[6]When comparing the estimated values of start of green to the observed ground-truth, we noticed that the error is inclined to the negative side. This is due to the value of a parameter called startup lost time ($t_{lost}$) which is the average time taken for a waiting bus to react to a signal changing to green. This lost time is used as follows to adjust the estimated start of green:

$$t_{start,adjusted} = t_{start} - t_{lost}$$

We varied the value of $t_{lost}$ to find a value that achieves the minimum RMS error in Fig. 14. We found that $t_{lost} = 6$ seconds results in minimum RMS error and included it in the results shown in Fig. 14 and in Table III

## XI. ACKNOWLEDGEMENT

This research was sponsored by a research award from BMW Group Technology Office, USA in Mountain View, California and by BMW Information Technology Research Center (ITRC) in Greenville, South Carolina. Alireza Fayazi was sponsored in part by the National Science Foundation grant number CMMI-0928533. The authors are thankful for the support provided by Mr. Hans-Peter Fischer from BMW ITRC. They also thank Mr. Michael Smith of NextBus for the data he provided. We thank Mr. Nianfeng Wan for his assistance with the Expectation Maximization algorithm.

## REFERENCES

[1] E. A. Mueller, "Aspects of history of traffic signals," *IEEE Transactions on Vehicular Technology*, vol. VT19, no. 1, pp. 6–17, 1970.
[2] 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*, vol. 19, no. 707-714, 2011.
[3] E. Koukoumidis, L.-S. Peh, and M. Martonosi, "Signalguru: Leveraging mobile phones for collaborative traffic signal schedule advisory," *Proceedings of MobiSys'11*, pp. 127–140, 2011.
[4] Department of Transportation, "Cooperative Intersection Collision Avoidance Systems," Web, http://www.its.dot.gov/cicas/cicas_overview.htm.
[5] 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 microcasm," in *Proceedings of AAAI Conference on Artificial Intelligence*, San Francisco, CA, 2011.
[6] National Transportation Operations Coalition, "National traffic signal report card," http://www.ite.org/REPORTCARD/.
[7] D. B. Work, O.-P. Tossavainen, S. Blandin, A. M. Bayen, T. Iwuchukwu, and K. Traction, "An ensemble kalman filtering approach to highway traffic estimation using gps enabled mobile devices," in *Proceedings of 47th Conference on Decision and Control*, Cancun, Mexico, 2008.
[8] J. C. Herrera, D. B Work, R. Herring, X. Ban, Q. Jacobson, and A. M. Bayen, "Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment," *Transportation Research part C*, vol. 18, pp. 568–583, 2010.
[9] A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen, "Learning the dynamics of arterial traffic from probe data using a dynamic bayesian network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, pp. 1679–1693, 2012.
[10] X. Ban, R. Herring, P. Hao, and A. Bayen, "Delay pattern estimation for signalized intersections using sample travel times.," *Transportation Research Record*, vol. 2130, pp. 109–119, 2009.
[11] M. Kerper, C. Wewetzer, A. Sasse, and M. Mauve, "Learning traffic light phase schedules from velocity profiles in the cloud," in *Proceedings of 5th International Conference on New Technologies, Mobility and Security (NTMS)*, 2012, pp. 1–5.
[12] Cabspotting, "http://cabspotting.org/," .
[13] Nextbus, "http://www.nextbus.com/," .
[14] J. L. Gattis, S. H. Nelson, and J.D. Tubbs, "School bus acceleration characteristics," Tech. Rep. FHWA/AR-009, Mack-Blackwell Transportation Center, University of Arkansas, 1998.
[15] S. Yoon, H. Li, J. Jun, J. Ogle, R. Guensler, and M. Rodgers, "A methodology for developing transit bus speed-acceleration matrices to be used in load-based mobile source emission models," in *Proceedings of TRB annual meeting*, 2005.
[16] G. Mahler and A. Vahidi, "Reducing idling at red lights based on probabilistic prediction of traffic signal timings," in *Proceedings of the American Control Conference*, Montreal, Quebec, 2012, pp. 6557–6562.
[17] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2007.
[18] Official United States Time by National Institutue of Standards and Technology, "http://nist.time.gov/," .

**S. Alireza Fayazi** is currently a Ph.D. student in mechanical engineering at Clemson University. He received his B.Sc. from K. N. Toosi University of Technology in 2005 in Iran, and his M.Sc. from University of Tehran in 2008, both in Electrical Engineering. He has been a visiting researcher at the University of California, Berkeley and has also been working as a visiting researcher in BMW Group Technology Office, USA in 2012-2013. Before joining Clemson University, he was a research engineer at Kerman Tablo Corp. for about three years where he was working on discrete control systems and digital control for embedded applications.

**Ardalan Vahidi** is currently an Associate Professor with the Department of Mechanical Engineering, Clemson University, Clemson, South Carolina. He received the Ph.D. degree in mechanical engineering from the University of Michigan, Ann Arbor, in 2005. the M.Sc. degree in transportation safety from George Washington University, Washington, DC, in 2002, and B.S. and M.Sc. degrees in civil engineering from Sharif University, Tehran, Iran, in 1996 and 1998, respectively. He has been a visiting scholar at the University of California, Berkeley and a visiting researcher at the BMW Group Technology Office USA in 2012-2013. His current research interests include control of vehicular and energy systems, and connected vehicle technologies.

**Grant Mahler** received his B.S. in mechanical engineering from Northwestern University in 2008. He is currently a Ph.D. candidate in mechanical engineering at Clemson University in Clemson, South Carolina. He was previously an intern at the BMW Information Technology Center in Munich, Germany, as well as a visiting scholar at the BMW Information Technology Research Center in Greenville, South Carolina. He is currently a visiting researcher at the BMW Group Technology Office USA in Mountain View, California, where his research focus remains on connected vehicle technologies.

**Andreas Winckler** is currently a Senior Advanced Technology Engineer and Project Leader at BMW Group Technology Office USA in Mountain View, California. He joined BMW in 2006 and worked on self learning navigation systems, map based driver assistance and predictive energy management systems before transferring to Mountain View in 2010 where he works on connected vehicle technologies. He was previously a Senior Systems Engineer at German Air Navigation Services and earned a Dipl.Ing. degree in aerospace engineering at University of Stuttgart in 1998.