Signal Timing Estimation Using Sample Intersection Travel Times

Peng Hao, Xuegang (Jeff) Ban, Kristin P. Bennett, Qiang Ji, *Senior Member, IEEE*, and Zhanbo Sun

***Abstract*—Signal timing information is important in signal operations and signal/arterial performance measurement. Such information, however, may not be available for wide areas. This imposes difficulty, particularly for real-time signal/arterial performance measurement and traffic information provisions that have received much attention recently. We study, in this paper, the possibility of using intersection travel times, i.e., those collected between upstream and downstream locations of an intersection, to estimate signal timing parameters. The method contains three steps: 1) cycle breaking that determines whether a new cycle starts; 2) exact cycle boundary detection that determines when exactly a cycle starts or ends; and 3) effective red (or green) time estimation that estimates the actual duration of the red (or green) time. The proposed method is a combination of traffic flow theory and learning/estimation algorithms and can be used to estimate the cycle-by-cycle signal timing parameters for a specific movement of a signal. The method is tested using data from microscopic simulation, field experiments, and next-generation simulation with promising results.**

***Index Terms*—Intersection travel times, mobile sensors, nonlinear programming, signal performance measurement, signal timing estimation, support vector machine (SVM).**

I. Introduction and Motivation

SIGNAL timing parameters, such as cycle length, number of phases, and effective red and green times of the phases, are important input to signal operations (such as signal coordination) and signal performance measurement. For example, existing methods for calculating steady-state signal performance measures, such as signal delays described in the Highway Capacity Manual [1], rely on static signal timing parameters, e.g., cycle length, and effective red and green times. Recently, estimating real-time signal performance measures, such as real-time delays or queue lengths, has received more attention [2]–[4]. While most of these methods used data from fixed location sensors (e.g., loop detectors) such as volume and occupancy, recent technological advances enable and promote the rapid deployment of sensing technologies to collect arterial intersection travel times directly. These technologies include, for example, electronic toll collection (ETC) readers, Bluetooth Mac address matching [5], wireless magnetic sensors [6], and mobile sensors such as Global Positioning System (GPS) cellular phones or other GPS devices [7], [8]. Ban *et al.* [4], [9] showed that intersection travel times can be used to estimate other real-time signal performance measures such as delay patterns, queue lengths, and even arrival volumes. Methods for real-time signal performance estimation, using either fixed location sensor data or intersection travel times, require both static and dynamic signal timing information, such as the cycle by cycle effective red (or green) times [4], [9].

In the current practice, collecting signal timing parameters directly from traffic signal controllers is probably trivial, particularly for small-scale data collection (such as for signals on a corridor or a small network). However, collecting such information for large areas (such as a region or nationwide) directly from controllers can be very challenging for several reasons. First, usually what is available is the signal timing plan sheet, which contains static signal timing information (e.g., cycle length, etc.) but cannot tell what will exactly happen in real time cycle by cycle; transportation agencies may not collect/archive at all those dynamic signal information. Second, signals are usually maintained and operated by multiple agencies under different data collection and monitoring systems—as a result, collecting wide-area signal timing information, particularly dynamic cycle-by-cycle information, is not a trivial task. Third, even if the dynamic signal information is collected and archived by each agency, it may not be easy for the agencies to release the data of large areas to a third party due to security and other related concerns.

On the other hand, leading traffic information providers have already started to provide wide-area real-time traffic information that aims to cover both freeways and arterials [10], [11]. The lack of wide-area dynamic signal timing information will likely limit the information provision for large-scale arterial networks. While it has been traditionally assumed that signal timing information should be available as an input to traffic models, given the current increasingly large amount of intersection travel time data, one intriguing question is can we instead infer static and dynamic signal timing parameters from intersection travel times, probably with the help of limited (and easily obtained) knowledge about the traffic signal? The answer to this question is not only scientifically interesting but is practically useful as well, particularly for traffic information providers who have already started to collect travel time information from various sources, as aforementioned. Based on the communications of the authors with researchers in the industry, they are anxious to integrate signal timing information with their arterial models. It seems that the most attractive way to do

Manuscript received May 9, 2011; revised August 9, 2011 and October 26, 2011; accepted December 3, 2011. Date of publication March 12, 2012; date of current version May 30, 2012. The work of P. Hao, X. Ban, and Z. Sun was supported by the U.S. National Science Foundation (NSF) under Grant CMMI-1031452. Any opinions, findings, and conclusions or recommendations in this paper are those of the authors and do not necessarily reflect the views of the NSF. The Associate Editor for this paper was W. Fan.

The authors are with the Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: haop@rpi.edu; banx@rpi.edu; bennek@rpi.edu; qji@ecse.rpi.edu; sunz2@rpi.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2012.2187895

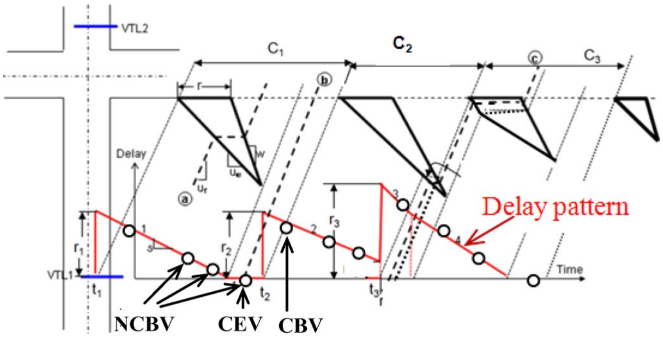


Fig. 1. Intersection delay pattern.

so is to infer signal timing information directly from the data that they have already collected, such as travel times. In this paper, we show that the answer to such a question is affirmative at least for signals with constant cycle lengths.

In fact, the authors in [9] showed preliminarily that signal timing parameters may be estimated using the intersection delay pattern. The delay pattern describes the delay that an imaginary vehicle may experience when passing the intersection at a certain time. It can be viewed as a continuous approximation of the measured (discrete) delays calculated from intersection travel times. Fig. 1 depicts the intersection delay pattern as a piecewise linear curve at the bottom of the figure. The small circles along the curve indicate the measured delays from individual vehicles. Here, travel times are collected between an upstream location (denoted as virtual trip line (VTL) 1) and a downstream location (denoted as VTL2). The VTL concept was initially developed for mobile sensors [7], which could also indicate the locations where ETC readers or wireless sensors are installed. The signal timing estimation method in [9] is based on the observation that a significant increase in the delay pattern is correlated with the start of the red time. By predefining a threshold for such an increase, the time that a new cycle starts can be detected. By further correlating the time when the delay pattern becomes zero with the duration of the red, the effective red time can also be estimated. The estimation method in [9], however, is not very satisfactory because of the following: 1) The method cannot very well determine the exact start/end times of a cycle; and 2) if the travel times are sparse, a significant delay increase may not exist for one or multiple consecutive cycles, and the method cannot detect those "missing" cycles.

In this paper, we develop a robust signal timing estimation method. The method features a combination of traffic flow theories and learning/optimization methods, which can estimate the exact cycle start/end times and can properly detect the missing cycles if any. The method contains three major steps: 1) cycle breaking estimation, which determine whether a new cycle starts; 2) exact cycle boundary detection, which detects the exact cycle start/end times; and 3) effective red (or green) time estimation, which estimates the duration of effective red (or green) times. Cycle breaking estimation applies the support vector machine (SVM) technique to classify sample vehicle delays into two groups, one of which indicates the start of red times. The classification results are more accurate and robust compared with the simple threshold method in [9]. Exact cycle boundary detection can be formulated as a nonlinear program by assuming that the cycle length is constant (the effective red and green times may vary from cycle to cycle, e.g., for the coordinated phases of actuated signals). A key feature of the exact cycle boundary estimation method is that the number of missing cycles can be estimated using sample delays and the SVM results. Finally, the effective red (or green) times are estimated using the method in [4] and [9] by investigating when nonsmoothness in the delay pattern happens.

The proposed signal timing estimation methods are tested in simulation and real-world data collected from next-generation simulation (NGSIM). The results are promising for relatively high penetration of travel time data (e.g., ≥10%–15%).

II. Robust Cycle Breaking Method

The first step for cycle parameter estimation is to detect when a cycle starts and ends. This is called "cycle breaking" in this paper. Here, we only illustrate how cycle breaking can be done for a particular movement of a signal, and the same procedure applies to other movements of the signal. In this paper, we define the start of a cycle as the start of the red time.

Cycle breaking can be done by exploring the correlation between the delay pattern of a signalized intersection and the start of the red time. This is because traffic at a signalized intersection has some periodic features due to signal timing. These features can be reflected by the measured intersection delays (or travel times) under relatively high penetration of mobile data. They can be seen via the discontinuities in the delay pattern in Fig. 1 when red times start. Usually, the vehicle that arrives at the end of a cycle, e.g., vehicle b, is not influenced by the signal and queues, whereas the vehicle that arrives at the beginning of the next cycle (during the red time) has to wait for the entire red time. On the other hand, if we know the delay pattern, the start of the red time can be inferred from the pattern by investigating when the discontinuity happens. For this purpose, we define a cycle breaking vehicle (CBV) as the first sample vehicle in a cycle. The other vehicles in this cycle are non-CBV (NCBV). CBV and NCBV are illustrated in Fig. 1. CEV in the figure stands for cycle ending vehicle, which will be defined later in this section. The CBV of a cycle is not necessarily the first vehicle actually arriving at the signal in the cycle if the penetration is not 100% (in this case, the first vehicle may not be sampled).

Generally, the delays of vehicles should be continuously decreasing within a cycle and "jumping" to a higher value when the next cycle starts. This implies that CBVs usually have higher delays. Such a feature can be used to detect whether a new cycle starts. In [9], e.g., a threshold is defined for this purpose. If the delay increase between one vehicle and the previous vehicle exceeds this threshold, a new cycle starts. However, using a single feature classifier is not robust if we consider oscillation and noise in measurements and, in particular, low penetration rate. To improve the performance, we use two features: the arrival time difference $t_i - t_{i-1}$ and the delay difference $d_i - d_{i-1}$ between two consecutively sampled vehicles. Here, t_i is the ith sample vehicle's arrival time at VTL1, and d_i is the intersection delay of the ith sample vehicle. Fig. 2 depicts these two features for a field experiment

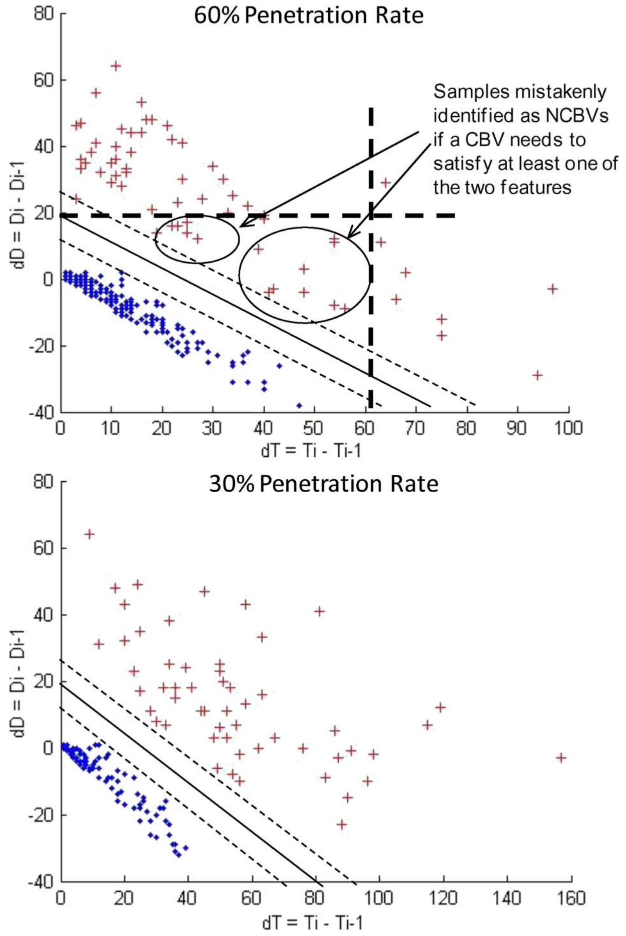


Fig. 2. Change in arrival time versus change in delay for different penetration rates.

conducted in the Albany, NY, area [4] under 60% and 30% penetration rates of travel time data. In the figure, dots are for NCBVs and plus signs are for CBVs. We can see that there is a clear margin of separation between CBVs and NCBVs using these two features. Notice that the margins (dashed lines) in Fig. 2 are drawn manually, purely based on the data and intuition for illustration purposes. In the numerical section of this paper, the margins are automatically generated.

The fact that CBVs and NCBVs can be separable using the two features intuitively makes sense. Since the first measure is the difference of the arrival times of two consecutively sampled vehicles, the larger this difference, the more likely that they are in two distinct cycles because the cycle length is finite. Since the second measure is the delay increase from the second vehicle to the first vehicle, a large value of this measure likely indicates the start of red time. The second measure is exactly what was used in [9] for cycle breaking. Using either feature or a simple combination of the two features, however, is not effective, as illustrated in Fig. 2. The vertical dashed bold line in the figure indicates the threshold in delay increase. The horizontal dashed bold line indicates the threshold in arrival times. The figure shows that even if both measures are used (e.g., a CBV needs to satisfy at least one of the two measures), there will still be large errors, as indicated by the circles.

In order to produce robust results, these two features need to be combined more intelligently for cycle breaking. Here, we use SVM, a widely used learning method, to classify a set of data points into two distinct groups [12]–[15]. Let the historical travel time data be denoted by $(\boldsymbol{x_i}, y_i)$, $i = 1, \ldots, M$ (in total M samples). Here, $\boldsymbol{x_i} = (t_i - t_{i-1}, d_i - d_{i-1})^T$ is a data point, and $y_i = \pm 1$ is the corresponding label ($y_i = 1$ for CBV and $y_i = -1$ for NCBV). SVM can help divide the data set into two groups: one for $y_i = 1$ and the other for $y_i = -1$ using two support planes (lines in the R^2 space, see Fig. 2). Let $\boldsymbol{w} = (w_1, w_2)^T \in R^2$ and b be a scalar. If $\boldsymbol{w}$ and b are properly selected, we will have $\boldsymbol{wx_i} - b \geq 1$ for $y_i = 1$ and $\boldsymbol{wx_i} - b \leq -1$ for $y_i = -1$. Then, the two support lines are $\boldsymbol{wx_i} - b = 1$ and $\boldsymbol{wx_i} - b = -1$. Notice that 1 and -1 can be used here by choosing proper scales for $\boldsymbol{w}$ and b. The distance between these two lines can be shown as $2/\|\boldsymbol{w}\|$ with $\|\boldsymbol{w}\|$, denoting the norm of $\boldsymbol{w}$. If we aim to maximize the distance between these two support planes, (w, b) can be determined by solving the following quadratic program [12]:

$$\min_{w,b} 1/2\|\boldsymbol{w}\|^2 \text{ s.t. } y_i(\boldsymbol{wx_i} - b) \geq 1, \quad i = 1, \ldots, M. \tag{1}$$

Notice that $y_i(\boldsymbol{wx_i} - b) \geq 1$ is a compact form for the two cases of $\boldsymbol{wx_i} - b \geq 1$ for $y_i = 1$ and $\boldsymbol{wx_i} - b \leq -1$ for $y_i = -1$. Model (1) is an SVM for separable cases, i.e., samples with $y_i = 1$ and $y_i = -1$ can be completely separated. The samples satisfying constraint (1) at equality are exactly on one of the two support planes—they are called support vectors. For many cases, there is no plane that can perfectly divide the two groups. In this case, we need to introduce an error term for each data point, denoted as ε_i. The problem is then to solve the following revised quadratic program [16]:

$$\min_{w,b} 1/2\|\boldsymbol{w}\|^2 + G\left(\sum_{i=1}^{M} \varepsilon_i\right)$$

$$\text{s.t. } y_i(\boldsymbol{wx_i} - b \geq 1 - \varepsilon_i, \varepsilon_i \geq 0, \qquad i = 1, \ldots, M. \tag{2}$$

The objective now is to minimize a weighted sum of the distance and the error term. Here, G is a weighting factor that can be defined by the user; a larger G assigns a larger penalty to the error.

We can solve the quadratic programming problem directly; the dual form of this problem is more preferable because its constraints are much simpler, i.e.,

$$\min_{a} \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} y_i y_j a_i a_j (\boldsymbol{x}_i \cdot \boldsymbol{x}_j) - \sum_{i=1}^{M} a_i$$

$$\text{s.t. } \sum_{i=1}^{M} y_i a_i = 0 \;\; G \geq a_i \geq 0. \;\; i = 1, \ldots, M. \tag{3}$$

The dual model (3) is a quadratic program of a. Given historical data $\{(\boldsymbol{x_i}, y_i)\}$, we can find the Lagrange multiplier $\{a_i\}$ by solving (3). The ith data point $(\boldsymbol{x_i}, y_i)$ is a support vector if its multiplier $a_i > 0$ and the normal to the plane can be calculated as $\sum_{i=1}^{M} y_i a_i \boldsymbol{x}_i$. Then, we have $y_i(\boldsymbol{wx_i} - b) = 1$ and $a_i > 0$ if and only if $(\boldsymbol{x_i}, y_i)$ is a support vector. This means that $b = \boldsymbol{wx_i} - (1/y_i)$ if the corresponding a_i is positive [15]. The middle plane $\boldsymbol{w} \cdot \boldsymbol{x_i} = b$ is then used to classify CBVs and

NCBVs. If $\boldsymbol{w} \cdot \boldsymbol{x_i} > b$ (or $w_1(t_i - t_{i-1}) + w_2(d_i - d_{i-1}) > b$), the ith sample vehicle is recognized as a CBV. If $\boldsymbol{w} \cdot \boldsymbol{x}_i \leq b$ (or $w_1(t_i - t_{i-1}) + w_2(d_i - d_{i-1}) \leq b$), the ith sample vehicle is classified as an NCBV. The CEV is defined as the sample vehicle that arrives just before a CBV. It is also the last sample vehicle in the previous cycle. The CEV in the nth cycle and the CBV in the $n+1$th cycle are called the nth cycle breaking pair. It is obvious that the cycle boundary between the nth cycle and the $n+1$th cycle should be somewhere between the times when these two vehicles arrived at the intersection. However, we do not know when exactly this happened. In [9], it was simply assumed that the cycle boundary is exactly at the middle point of the two arrival times. We show next how the exact cycle boundaries can be detected based on some engineering knowledge about the signal.

III. Exact Cycle Boundary Estimation

To detect the exact boundaries of cycles, further knowledge about the signal is needed. This is where engineering knowledge can play a critical role in developing specialized learning and optimization methods. Next, we show how exact cycle boundaries can be determined if the signal has a constant cycle length. This applies to pretimed signals or the coordinated phases of coordinated actuated signals. For pretimed signals, the cycle length and red/green times are fixed. For coordinated and actuated signals, we have a fixed cycle length but variable red/green times for the coordinated phase. To detect exact cycle boundaries, we need to first figure out the exact times when CBV and CEV arrive at the intersection (i.e., the stop line).

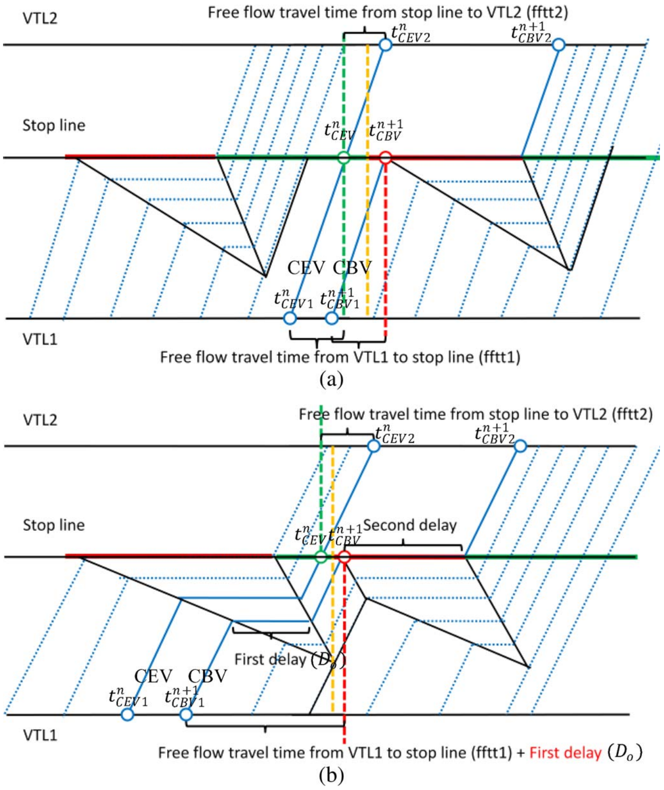
Here, we discuss normal cycles, i.e., the queue can be fully discharged during the green time of the cycle, and oversaturated cycles, i.e., the queue cannot be fully discharged during the green time of a cycle and some vehicles have to wait for an extra red time. To simplify the discussion, we assume that an oversaturated vehicle waits only one extra red time to get through the intersection. See [4] for more details of those definitions.

A. Arrival Time at the Stop Line

As defined in the previous section, CBV is the first sample vehicle within one cycle. If the penetration rate is 100%, each CBV should be the first vehicle that actually arrives at the stop line after the start of red time; each CEV should be the last vehicle that actually passes the stop line before the start of red time. Because we can only detect the arrival times at VTL1 and VTL2, we have to process them into times at the stop line. For different traffic conditions, we have different methods to compute the arrival times at the stop line.

For normal conditions in Fig. 3(a), we show the trajectories of CEV and CBV using solid lines and the other vehicles using dashed lines. The time that CEV passes the stop line (denoted as t^n_{CEV}) can be expressed as the difference between the arrival time at VTL2 (denoted as t^n_{CEV2}) and the free flow travel time from stop line to VTL2 (denoted as $fftt2$)

$$t^n_{\mathrm{CEV}} = t^n_{\mathrm{CEV2}} - fftt2. \tag{4}$$



(a)

(b)

Fig. 3. Arrival time at the stop line. (a) Normal condition. (b) Oversaturation condition.

The time that CBV arrives at the stop line (denoted as t^{n+1}_{CBV}) can be expressed as the difference between the arrival time at VTL1 (denoted as t^{n+1}_{CBV1}) and the free flow travel time from VTL1 to stop line (denoted as $fftt1$)

$$t^{n+1}_{\mathrm{CBV}} = t^{n+1}_{\mathrm{CBV1}} + fftt1. \tag{5}$$

If the first vehicle in a cycle is not sampled, i.e., if CBV is not the first vehicle actually in the queue, t^{n+1}_{CBV} derived from (5) will be the time that a queued vehicle stops at the stop line, which still provides an upper bound for the boundary.

For oversaturation conditions, the time that the CEV passes the stop line can still be expressed by (4), as shown in Fig. 3(b). This is because the CEV must not be oversaturated by definition. However, the CBV can be an oversaturated vehicle that stops twice in front of the stop line so that (5) cannot be applied to compute t^{n+1}_{CBV}. We have to consider the first delay D_0 defined as the delay of the oversaturated vehicle if there were enough green time in the nth cycle, as shown in Fig. 3(b). It is also the delay an oversaturated vehicle experienced during the first cycle. In [9], we proposed a piecewise linear delay model to calculate the intersection delay over time. If the green time of the nth cycle is long enough, all vehicles arriving during the nth cycle follow the same delay reduction function, for example, $D = \alpha_0 - \alpha_1 t$. Here, t is the vehicle arrival time at VTL1, and the positive coefficients α_0 and α_1 are parameterized by linear fitting (for details, see [9]). Thus, we estimate the first delay of the CBV by the following equation:

$$\overline{D}_0 = \alpha_0 - \alpha_1 t^{n+1}_{\mathrm{CBV1}} \tag{6}$$

where $\overline{D}_0$ is the estimated value of the first delay for CBV. If the $\overline{D}_0$ of the CBV is positive, we can conclude that this cycle is under oversaturation and derive t_{CBV}^{n+1} by

$$t_{\mathrm{CBV}}^{n+1} = t_{\mathrm{CBV1}}^{n+1} + fftt1 + \overline{D}_0. \tag{7}$$

If $\overline{D}_0$ of the CBV is negative or zero, we say either of the following: 1) The cycle is under normal condition; or 2) the cycle is under oversaturation, but no oversaturated vehicle is sampled (e.g., due to low penetration). In either case, we can calculate t_{CBV}^{n+1} by (5) because that particular CBV is a normal vehicle.

Notice here that the derivations of t_{CBV}^{n+1} and t_{CEV}^{n} assume a uniform arrival pattern of the intersection, which may not be true in reality. However, since t_{CBV}^{n+1} and t_{CEV}^{n} are only used in this paper to provide upper and lower bounds of the exact cycle start/end times, the numerical results in Section V show that such an assumption works reasonably well. For more detailed discussions of the limitations of the uniform arrival assumption, one can refer to [4]. In [17], the authors also proposed a method to partially relax the uniform arrival assumption when estimating real-time intersection queue length.

B. Exact Cycle Boundary Estimation Without Missing Cycles

For signals with constant cycle lengths, it is possible to estimate the exact boundaries of cycles by formulating a nonlinear programming model. The constant cycle length can be considered as a constraint of the model. The key of the model is to observe that the boundary of the nth and $n+1$th cycles should be located between the arrival times, at the stop line, of the cycle breaking pair, i.e., the CEV in the nth cycle and the CBV in the $n+1$th cycle. This way, each cycle breaking pair provides a constraint to the problem. To have an unbiased estimate, we assume that the objective is to minimize the deviation of the boundary from the middle point of the arrival times of CEV and CBV. This leads to the following basic model for cycle boundary detection.

Basic Model:

$$\begin{aligned} &\min_{t_0,C} \frac{1}{N}\sum_{n=1}^{N}\left(t_0 + nC - \frac{t_{\mathrm{CEV}}^{n} + t_{\mathrm{CBV}}^{n+1}}{2}\right)^2 \\ &\text{s.t. } t_0 + nC \geq t_{\mathrm{CEV}}^{n}, \quad n = 1,2,\ldots,N \\ &\qquad t_0 + nC \leq t_{\mathrm{CBV}}^{n+1}, \quad n = 1,2,\ldots,N. \end{aligned} \tag{8}$$

Here, t_0 is the start of the red time of the first cycle, C is the fixed cycle length, and N is the number of cycle breaking pairs, indicating that the number of cycles in the basic model is $N+1$. Then, $t_0 + nC$ is the end of the nth cycle (or the start of the $n+1$th cycle). By solving this nonlinear programming problem, we can obtain the estimated cycle length and the start of the red time of each cycle. Notice that the basic model (8) and the other models developed in this section are all convex quadratic programs that are fairly easy to solve. We can also easily see that the cycle boundary $t_0 + nC = (t_{\mathrm{CEV}}^{n} + t_{\mathrm{CBV}}^{n+1}/2)$ will be a global solution if the problem (8) is feasible. This indicates that the heuristics developed in [9] works well only when (8) is feasible.

Due to data errors, it is possible that model (8) does not have any feasible solution (which actually happens for most testing cases in Section V). In this case, it is necessary to introduce error terms on the boundary constraints. This results in an extended model for cycle boundary detection.

Extended Model:

$$\begin{aligned} &\min_{t_0,C,\varepsilon} \frac{1}{N}\sum_{n=1}^{N}\left(t_0 + nC - \frac{t_{\mathrm{CEV}}^{n} + t_{\mathrm{CBV}}^{n+1}}{2}\right)^2 + \frac{K}{N}\sum_{i=n}^{N}\varepsilon_n^2 \\ &\text{s.t. } t_0 + nC \geq t_{\mathrm{CEV}}^{n} - \varepsilon_n, \quad n = 1,2,\ldots,N \\ &\qquad t_0 + nC \leq t_{\mathrm{CBV}}^{n+1} + \varepsilon_n, \quad n = 1,2,\ldots,N. \end{aligned} \tag{9}$$

The error variable ε_i denotes the tolerance on how far the cycle boundary can deviate from the interval defined by the arrival times between the CEV (of the nth cycle) and the CBV (of the $n+1$th cycle). We also introduce a weighted penalty term $K > 0$ for the error terms. In this paper, K is chosen the same as N based on some numerical experiments on different choices.

The cycle boundary models (8) and (9) rely on two assumptions: 1) The SVM model developed previously produced satisfactorily accurate results in terms of identifying CEVs and CBVs, and 2) the CEV and CBV can be detected for each and every cycle. Assumption 1 depends on the penetration rate of sample travel times. We will show later that beyond certain penetration, SVM can always produce highly accurate results. Assumption 2, however, may not be valid, even under high penetration due to the variation of traffic. It is thus crucial to identify whether there are any missing cycles between a CEV and a CBV and, if yes, how many of them are missing, to apply the model (8) or (9) properly.

C. Missing Cycle Identification

Under a low penetration rate, it is possible that no vehicle in a cycle can be detected. In this case, there will be no CEV or CBV in the cycle. Sometimes, we have even more than one cycle missing between a CEV and the next CBV. It is thus important to detect missing cycles and identify the number of missing cycles before (8) or (9) can be applied.

First, we focus on the case when the cycle length C is known. Between the nth cycle breaking pairs, denote m_n as the number of missing cycles. Assume that T_r^n is the actual start of the red time in the nth detected cycle (which is before t_{CEV}^{n}). Then, the actual start time of the red time of the $n+1$th detected cycle is

$$T_r^{n+1} = T_r^n + C(m_n + 1). \tag{10-a}$$

Notice here that "n" and "$n+1$" are detected cycles from the cycle breaking pair, which are not exactly the cycle indices that actually happened if there are missing cycles between the cycle breaking pairs. Then, t_{CEV}^{n}, the arrival time of the CEV in the nth detected cycle, is between the start and end

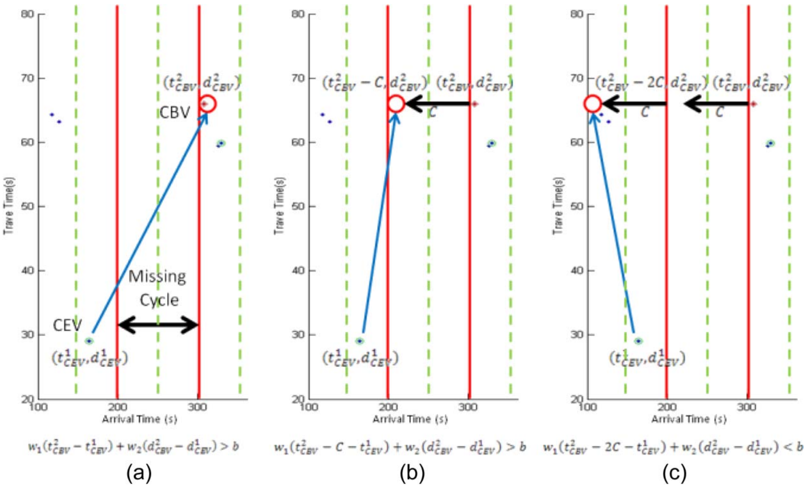


Fig. 4. Missing cycle identification process. (a) Cycle breaking. (b) Missing cycle ≥ 1. (c) Missing cycle < 2.

times of the nth detected cycle, i.e., we have $T_r^n \leq t^n_{\text{CEV}} < T_r^n + C$. Similarly, the arrival time of the CBV in the $n+1$th detected cycle is between the start and end times of the $n+1$th detected cycle, which satisfies $T_r^{n+1} \leq t^{n+1}_{\text{CBV}} < T_r^{n+1} + C$. This leads to

$$T_r^n + C(m_n + 1) \leq t^{n+1}_{\text{CBV}} < T_r^n + C(m_n + 2). \tag{10-b}$$

If we move the CBV point $(t^{n+1}_{\text{CBV}}, d^{n+1}_{\text{CBV}})$ m_n times cycle lengths (i.e., Cm_n) to the left, the new point $(t^{n+1}_{\text{CBV}} - m_n C, d^{n+1}_{\text{CBV}})$ satisfies $T_r^n + C \leq t^{n+1}_{\text{CBV}} - m_n C < T_r^n + 2C$. Because this new CBV is in the cycle right after the CEV's cycle, the support vectors obtained via solving the SVM model in Section II should detect this CBV as indeed being a CBV (they are at two different cycles). In other words, we have

$$w_1\left(t^{n+1}_{\text{CBV}} - m_n C - t^n_{\text{CEV}}\right) + w_2\left(d^{n+1}_{\text{CBV}} - d^n_{\text{CEV}}\right) > b. \tag{11}$$

Here, w_1, w_2, and b are obtained by solving the SVM model (see Section II). If we move the CBV point $(m_n + 1)$ cycle lengths to the left, the new point $(t^{n+1}_{\text{CBV}} - (m_n + 1)C, d^{n+1}_{\text{CBV}}))$ satisfies $T_r^n \leq t^{n+1}_{\text{CBV}} - (m_n + 1)C \leq T_r^n + C$. Because this new CBV is in the nth detected cycle, the same cycle as the CEV, SVM should identify the new vehicle as an NCBV. Thus

$$w_1\left[t^{n+1}_{\text{CBV}} - (m_n + 1)C - t^n_{\text{CEV}}\right] + w_2\left(d^{n+1}_{\text{CBV}} - d^n_{\text{CEV}}\right) \leq b. \tag{12}$$

The number of missing cycles m_n can then be derived by (11) and (12) as

$$\begin{aligned} &\frac{w_1\left(t^{n+1}_{\text{CBV}} - t^n_{\text{CEV}}\right) + w_2\left(d^{n+1}_{\text{CBV}} - d^n_{\text{CEV}}\right) - b}{w_1 C} - 1 \leq m_n \\ &\qquad < \frac{w_1\left(t^{n+1}_{\text{CBV}} - t^n_{\text{CEV}}\right) + w_2\left(d^{n+1}_{\text{CBV}} - d^n_{\text{CEV}}\right) - b}{w_1 C}. \end{aligned} \tag{13}$$

Since m_n is an integer, we can use the following equation:

$$m_n = \left\lfloor \frac{w_1\left(t^{n+1}_{\text{CBV}} - t^n_{\text{CEV}}\right) + w_2\left(d^{n+1}_{\text{CBV}} - d^n_{\text{CEV}}\right) - b}{w_1 C} \right\rfloor. \tag{14}$$

Here, the notation $\lfloor x \rfloor$ represents the integer part of x, which is widely known as the floor function. Obviously, m_n is a monotonic nonincreasing function in terms of C.

Fig. 4 illustrates the missing cycle identification process when $m_n = 1$. In the figure, the start times of red are plotted using vertical solid lines, and the start times of green are plotted using vertical dashed lines. In Fig. 4(a), we find a cycle breaking pair via SVM. Obviously, there is one missing cycle between them. Then, we move the CBV point $(t^2_{\text{CBV}}, d^2_{\text{CBV}})$ to the left for one cycle in Fig. 4(b) and two cycles in Fig. 4(c). Then, (11) should apply for Fig. 4(b) and (12) should apply for Fig. 4(c). That is

$$w_1\left(t^2_{\text{CBV}} - C - t^1_{\text{CEV}}\right) + w_2\left(d^2_{\text{CBV}} - d^1_{\text{CEV}}\right) > b$$

$$w_1\left(t^2_{\text{CBV}} - 2C - t^1_{\text{CEV}}\right) + w_2\left(d^2_{\text{CBV}} - d^1_{\text{CEV}}\right) < b.$$

Hence, we conclude that the missing cycle number $m_n = 1$ from (14).

For many cases, the cycle length C is unknown. If we know the upper bound C_u and the lower bound C_l of the cycle length, m_n should be restricted by the minimum missing cycle number $\underline{m}_n$ and the maximum missing cycle number $\overline{m}_n$, i.e., $\underline{m}_n \leq m_n < \overline{m}_n$, where

$$\underline{m}_n = \left\lfloor \frac{w_1\left(t^{n+1}_{\text{CBV}} - t^n_{\text{CEV}}\right) + w_2\left(d^{n+1}_{\text{CBV}} - d^n_{\text{CEV}}\right) - b}{w_1 C_u} \right\rfloor \tag{15}$$

$$\overline{m}_n = \left\lfloor \frac{w_1\left(t^{n+1}_{\text{CBV}} - t^n_{\text{CEV}}\right) + w_2\left(d^{n+1}_{\text{CBV}} - d^n_{\text{CEV}}\right) - b}{w_1 C_1} \right\rfloor. \tag{16}$$

In practice, C_u and C_l can be estimated via engineering knowledge. For example, the cycle lengths of most signals in the US are between 30 and 180 s. On the other hand, C_u and C_l can be better estimated using sample arrival times and the estimated number of missing cycles. In other words, we can compute the estimated cycle length and the number of missing cycles iteratively to refine the estimation results. This procedure can be described as follows.

At the beginning of the iteration, we define the initial state

$$C_l^0 = 0,\ C_u^0 = \infty,\ \underline{m}_n = 0,\ \overline{m}_n = \infty,\ n = 1, 2, 3, \ldots N.$$

In the jth iteration, we check the arrival time difference between CEV in the nth detected cycle and CBV in the $(n+k-1)$th detected cycle. When $k = 1$, two vehicles are in the same cycle, i.e.,

$$C \geq t_{\mathrm{CEV}}^n - t_{\mathrm{CBV}}^n \quad n = 1, 2, 3, \ldots N. \tag{17}$$

When $k > 1$, because $t_{\mathrm{CEV}}^{n+k-1} \leq T_r^{n+k-1} + C$ and $t_{\mathrm{CBV}}^n \geq T_r^n$

$$\begin{aligned} t_{\mathrm{CEV}}^{n+k-1} - t_{\mathrm{CBV}}^n &\leq T_r^{n+k-1} + C - T_r^n \\ &= T_r^n + (k-1)C + C \sum_{i=n}^{n+k-2} m_i + C - T_r^n \\ &= kC + C \sum_{i=n}^{n+k-2} m_i,\ k = 2, 3, 4, \ldots \\ \Rightarrow C &\geq \frac{t_{\mathrm{CEV}}^{n+k-1} - t_{\mathrm{CBV}}^n}{k + \sum_{i=n}^{n+k-2} m_i} \geq \frac{t_{\mathrm{CEV}}^{n+k-1} - t_{\mathrm{CBV}}^n}{k + \sum_{i=n}^{n+k-2} \overline{m}_i^{j-1}} \\ & n = 1, 2, \ldots N-1,\ k = 2, 3, \ldots N-n+1. \end{aligned} \tag{18}$$

The lower bound of cycle length in the jth iteration should be the maximum value among $t_{\mathrm{CEV}}^n - t_{\mathrm{CBV}}^n$ [from (17)] and all the $(t_{\mathrm{CEV}}^{n+k-1} - t_{\mathrm{CBV}}^n / k + \sum_{i=n}^{n+k-2} \overline{m}_i^{j-1})$ for different k's. Thus, mathematically, the lower bound of the cycle length at the jth iteration $C_i^j = max(S_i^j)$, where

$$\begin{aligned} S_i^j = &\{t_{\mathrm{CEV}}^n - t_{\mathrm{CBV}}^n : n = 1, 2, \ldots N\} \\ &\cup \left\{ \frac{t_{\mathrm{CEV}}^{n+k-1} - t_{\mathrm{CBV}}^n}{k + \sum_{i=n}^{n+k-2} \overline{m}^{j-1}} \right. \\ &\left. : n = 1, 2 \ldots N-1,\ k = 2, 3 \ldots N-n+1 \right\}. \end{aligned}$$

Note that in the first iteration $\overline{m}_n^0 = \infty$. $(t_{\mathrm{CEV}}^{n+k-1} - t_{\mathrm{CBV}}^n / k + \sum_{i=n}^{n+k-2} \overline{m}_i^{j-1}) = 0$ for each k. The lower bound of cycle length C_l^0 is $max\{t_{\mathrm{CEV}}^n - t_{\mathrm{CBV}}^n : n = 1, 2 \ldots N\}$.

The maximum number of missing cycles in the jth iteration $\overline{m}_i^j$ then should be

$$\overline{m}_n^j = \left\lfloor \frac{w_1 \left(t_{\mathrm{CBV}}^{n+1} - t_{\mathrm{CEV}}^n\right) + w_2 \left(d_{\mathrm{CBV}}^{n+1} - d_{\mathrm{CEV}}^n\right) - b}{w_1 C_l^j} \right\rfloor. \tag{19}$$

Similarly, we check the arrival time difference between the CEV in the nth detected cycle and the CBV in the $(n+k-1)$th detected cycle. Because $t_{\mathrm{CEV}}^n \leq T_r^n + C$ and $t_{\mathrm{CBV}}^{n+k+1} \geq T_r^{n+k+1}$

$$\begin{aligned} t_{\mathrm{CBV}}^{n+k+1} - t_{\mathrm{CEV}}^n &\geq T_r^{n+k+1} - T_r^n - \\ &= T_r^n + (k+1)C + C \sum_{i=n}^{n+k-2} m_i - T_r^n \\ &= kC + C \sum_{i=n}^{n+k} m_i, k = 1, 2, 3, 4, \ldots \\ C &\leq \frac{t_{\mathrm{CBV}}^{n+k+1} - t_{\mathrm{CEV}}^n}{k + \sum_{i=n}^{n+k} m_i} \leq \frac{t_{\mathrm{CBV}}^{n+k+1} - t_{\mathrm{CEV}}^n}{k + \sum_{i=n}^{n+k} \underline{m}_i^{j-1}} \\ N &= 1, 2, \ldots N-2,\ k = 1, 2, \ldots N-n. \end{aligned} \tag{20}$$

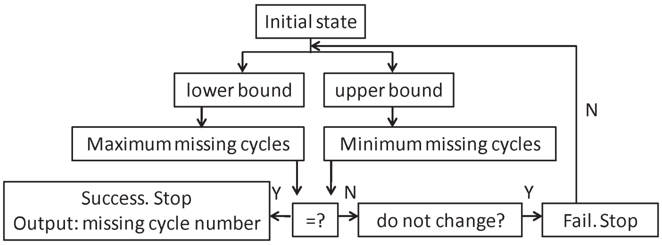


Fig. 5. Flowchart of missing cycle detection algorithm.

The upper bound of the cycle length in the jth iteration should be the minimum value among all the $(t_{\mathrm{CBV}}^{n+k+1} - t_{\mathrm{CEV}}^n / k + \sum_{i=n}^{n+k} \underline{m}_i^{j-1})$ for different k, i.e., $C_u^j = max(S_u^j)$, where

$$S_u^j = \left\{ \frac{t_{\mathrm{CBV}}^{n+k+1} - t_{\mathrm{CEV}}^n}{k + \sum_{i=1}^{n+k} \underline{m}_i^{j-1}} : \right. \left. n = 1, 2, \ldots N-2,\ k{=}1, 2, \ldots N-n \right\}.$$

The minimum number of missing cycles in the jth iteration $\underline{m}_i^j$ then should be

$$\underline{m}_n^j = \left\lfloor \frac{w_1 \left(t_{\mathrm{CBV}}^{n+1} - t_{\mathrm{CEV}}^n\right) + w_2 \left(d_{\mathrm{CBV}}^{n+1} - d_{\mathrm{CEV}}^n\right) - b}{w_1 C_u^j} \right\rfloor. \tag{21}$$

We repeat the process until one of the two conditions holds.

a) The minimum number of missing cycles is equal to the maximum number of missing cycles between each cycle breaking pair;
b) The minimum number of missing cycles is not equal to the maximum number of missing cycles, but neither of them changes when the iteration proceeds.

For the first case, the minimum and maximum numbers of missing cycles converge to the true value. We can derive the number of missing cycles successfully and then apply the cycle boundary detection model in the next section. For the second case, the exact number of missing cycles cannot be determined when the algorithm is terminated. This case only occurs when the penetration rate is very low. We will claim this case as a failure in the numerical test in Section V. The process of missing cycle detection can be illustrated using the flowchart in Fig. 5.

D. Exact Cycle Boundary Estimation With Missing Cycles

After the number of missing cycles between each cycle breaking pair is known, we can further improve model (9). First, define E_n and $\bar{t}_{\text{CBV}}^{n+1}$ as follows:

$$E_n = n + \sum_{i=1}^{n-1} m_i \tag{22-a}$$

$$\bar{t}_{\text{CBV}}^{n+1} = t_{\text{CBV}}^{n+1} - Cm_n. \tag{22-b}$$

Note that E_n and $\bar{t}_{\text{CBV}}^{n+1}$ can readily be calculated if the number of missing cycles between each cycle breaking pair is known. Then, $t_0 + E_nC$ is the end of the nth detected cycle, bounded by t_{CBV}^{n} and t_{CBV}^{n+1}. If we remove all missing cycles between the nth and $n+1$th detected cycles, the arrival time of the new CBV $\bar{t}_{\text{CBV}}^{n+1}$ should still be after the end of nth detected cycle. This leads to the following generalized model.

General Model:

$$\begin{aligned}\min_{t_0,C,\varepsilon} \frac{1}{N}\sum_{n=1}^{N}\left[t_0 + E_nC - \frac{t_{\text{CEV}}^{n} + \bar{t}_{\text{CBV}}^{n+1}}{2}\right] + \frac{K}{N}\sum_{n=1}^{N}\varepsilon_n^2 \\ \text{s.t.}\quad t_0 + E_nC \geq t_{\text{CEV}}^{n} - \varepsilon_n\ \forall n \\ t_0 + E_nC \leq \bar{t}_{\text{CBV}}^{n+1} + \varepsilon_n\ \forall n.\end{aligned} \tag{23}$$

The objective in (23) is to minimize the deviation from the end of the nth detected cycle to the middle point of t_{CEV}^{n} and $\bar{t}_{\text{CBV}}^{n+1}$ (i.e., $t_{\text{CBV}}^{n+1} - Cm_n$). The constraints are similar to those in (9). Here, (23) is in exactly the same form compared to (9) by replacing nC and t_{CBV}^{n+1} in (9) with E_nC and $\bar{t}_{\text{CBV}}^{n+1}$, respectively, in (23). Solving the foregoing nonlinear programming model, we can find the estimated start of the first cycle t_0 and cycle length C. Notice here that the construction of the general model (23) relies on the identification of missing cycles, which is based on engineering knowledge (about signal and delay characteristics) and the results of the SVM [e.g., (11) and (12)]. One might also be able to integrate the missing cycle identification step directly into the optimization/learning model as a pure (but likely more complex and challenging to solve) optimization/learning problem without using the foregoing engineering-knowledge-based missing-cycle identification procedure. However, (23) is obviously much simpler and easier to solve. This shows the benefits of combining engineering knowledge and optimization/learning techniques in the modeling procedure to solve the proposed traffic problem.

IV. Effective Red Time Estimation

The effective red times can be estimated using the signal timing estimation method in [9]. For normal conditions, the duration of the red time is equal to the estimated delay at the start of the red time. For oversaturation conditions, the duration of the red time is equal to the estimated delay at the start of the red minus the estimated first delay defined in Section III. The duration of the effective green can then be derived if the effective red time and cycle length are known. For more detailed descriptions, see [9]. We then summarize the signal timing parameter estimation algorithm as follows.

Signal Timing Estimation Algorithm

Step 1: Cycle breaking estimation. Calculate changes in arrival times and changes in delays of any two consecutive sampled vehicles. Find CEVs and CBVs by using the SVM.

Step 2: Oversaturation identification. Calculate the first delay using (6). Identify oversaturation if the estimated first delay is positive. Otherwise, the cycle is under normal condition.

Step 3: Arrival time calculation. Compute the arrival times of CEVs and CBVs at the stop line for both normal and oversaturation conditions using (4), (5), and (7).

Step 4: Missing cycle detection. Follow the procedure in Fig. 5 and repeat until the minimum and maximum numbers of missing cycles converge or do not change. If the two numbers converge, go to step 5; otherwise, go to step 7.

Step 5: Exact cycle boundary estimation. Estimate the initial start of red, cycle length, and exact cycle boundaries using the general model (23).

Step 6: Effective red and green times. The effective red and green times can be estimated by the method in [9]. End.

Step 7: Failure. Claim that the algorithm failed to estimate signal timing parameters.

V. Numerical Experiments

We test the signal timing estimation method in this paper using multiple data sets, including microscopic simulation, field experiment, and NGSIM. The simulation model was developed using Paramics (a commercial traffic simulator) for the City of Fresno, CA [18], [19]; the field test was conducted in Albany, NY, for the left turn movement of an actuated (but uncoordinated) intersection [4]; NGSIM data were collected at the Peachtree St in Atlanta, GA, covering four pretimed intersections [20]. We use the two data sets collected for the intersection of 14th St NE and Peachtree St: one from 12:45 to 1:00 P.M. and the other from 4:00 to 4:15 P.M.. Since the field test intersection does not have a constant cycle length, the proposed exact cycle boundary method in this paper cannot apply. We thus only apply this data set to test the cycle breaking method. The NGSIM and simulation data are used for all three steps: 1) cycle breaking estimation; 2) exact cycle boundary detection; and 3) effective red (green) time estimation.

A. Cycle Breaking Estimation

Fig. 6(a)–(k) shows the cycle breaking results for the field test, NGSIM data, and data sets for nine intersections in simulation. For the field test data, we use the first 15-min data for training and the remaining 45 min for testing. Since the two NGSIM data sets are only 15 min each, we use the data set from 12:45 to 1:00 P.M. as the training data set for SVM and the data set from 4:00 to 4:15 P.M. as the testing data set. In the simulation, we ran the simulation for 1 h and used the first 15 min as the training data and the remaining 45 min as the testing data. The penetration rate for the results in Fig. 6 is 30%. We can see that the proposed SVM-based cycle breaking

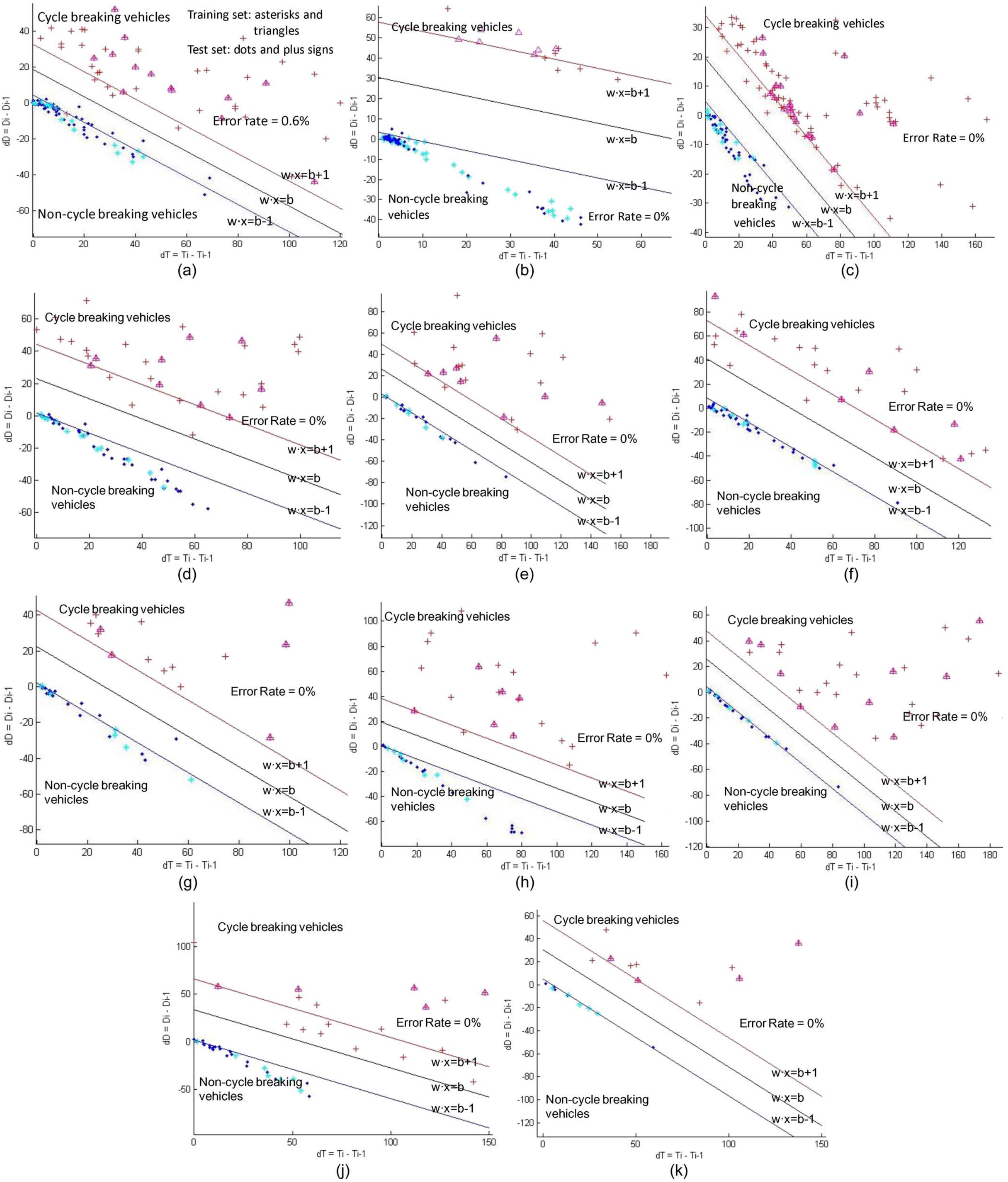


Fig. 6. Cycle breaking results. (a) Field test data, uncoordinated phase of actuated signal. (b) NGSIM data, pretimed signal. (c) Simulation 1, pretimed signal. (d) Simulation 2, uncoordinated phase of actuated signal. (e) Simulation 3, uncoordinated phase of actuated signal. (f) Simulation 4, uncoordinated phase of actuated signal. (g) Simulation 5, uncoordinated phase of actuated signal. (h) Simulation 6, uncoordinated phase of actuated signal. (i) Simulation 7, uncoordinated phase of actuated signal. (j) Simulation 8, uncoordinated phase of actuated signal. (k) Simulation 9, uncoordinated phase of actuated signal.

method can almost perfectly identify CBVs and NCBVs under this penetration rate; there is only one mismatched vehicle (out of about 150 vehicles) for the field test data. We also tested the method for other penetration rates (as low as 5%) and other data sets in simulation. Similar results have been obtained, and we found that the cycle breaking method can generate a very

TABLE I
PERFORMANCE OF SIGNAL TIMING ESTIMATION

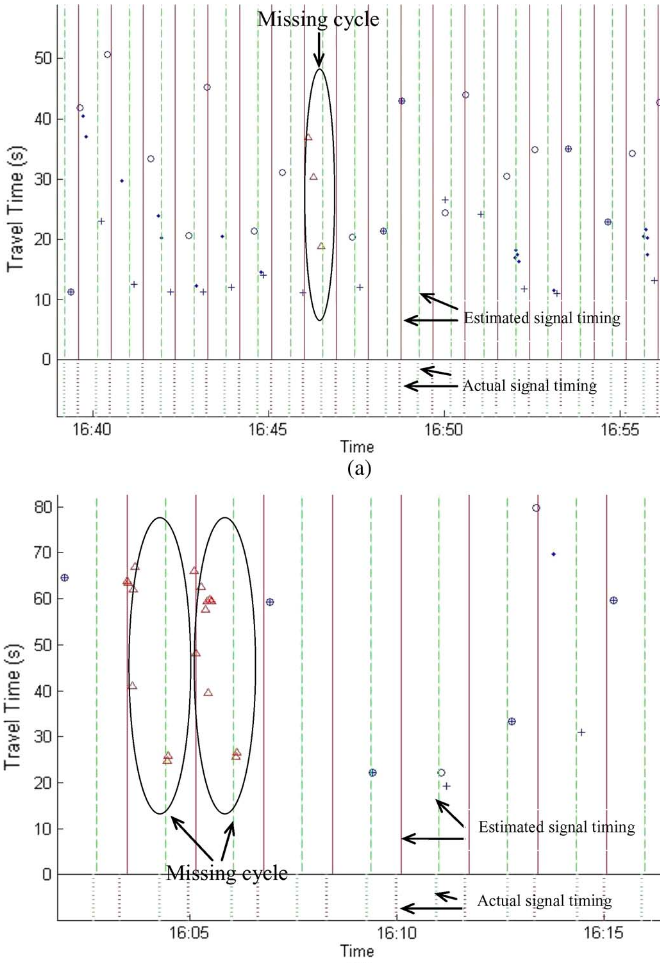
Simulation			
	Observation	Estimation	Absolute error
Start of red time in the first cycle (s)	55904.250	55905.720	1.470
Cycle length (s)	55.000	55.003	0.003
Effective red time length (s)	31.000	30.210	0.790
RMSE by the proposed method (s)	1.632		
RMSE by method of [9] (s)	9.593		
NGSIM			
	Observation	Estimation	Absolute error
Start of red time in the first cycle (s)	98.000	99.916	1.916
Cycle length (s)	100.000	100.370	0.037
Effective red time length (s)	68.000	70.210	2.210
RMSE by the proposed method (s)	3.077		
RMSE by method of [9] (s)	8.249		

accurate classification result when the penetration is relatively high (e.g., >=10%–15%). The results also indicate that the cycle breaking method can apply to signals with either fixed cycle length (e.g., pretimed signals or coordinated phases of an actuated signal) or variable cycle length (e.g., for uncoordinated phases of an actuated signal). The results clearly show that CBVs and NCBVs are linearly separable by the two measures used in the SVM: 1) arrival time difference and 2) delay difference of two consecutively sampled vehicles (see Section II). This explains why the proposed linear SVM is effective.

B. Exact Cycle Boundary and Effective Red Time Estimation

The exact cycle boundary detection and effective red/green time estimation are tested using two pretimed intersections: the first intersection in simulation [see Fig. 6(c)] and the NGSIM data, as shown in Table I. For the simulation data, there is only 1.47-s deviation in the estimated start of red time in the first cycle. The estimated and observed cycle lengths are almost the same. We can also compute the effective red time using line fitting, i.e., step 6 of the algorithm in Section IV. The root mean square error (RMSE) between the observed and estimated start of red times for all the cycles is 1.632 s. We also calculated the signal timing parameters using the method in [9], which has an RMSE of 9.593 s. For the NGSIM data, the deviation in the estimated start of red time of the first cycle is 1.9 s. The estimated and observed cycle lengths are almost the same: 100.0 and 100.37, respectively. The RMSE between the observed and estimated start of red times for all the nine cycles is about 3 s, which is significantly smaller than that by the method in [9], which is more than 8 s. Table I shows that the method proposed in this paper greatly outperforms the method in [9].

Fig. 7(a) shows the estimation of the exact cycle boundaries for the simulation data for 50% penetration. The estimated start times of red are plotted using vertical solid lines, and the estimated start times of green are plotted using vertical dashed lines. The actual starts of red and green times are also shown in the figure using dotted lines. We depict the travel times of CEVs by circles, CBVs by plus signs, and other vehicles by dots. Note that if there is only one vehicle in the cycle, it is considered as both CEV and CBV. We can see that the samples of travel times are well split by the estimated cycle boundaries. The estimated starts of red and green times match very well with the actual starts of the red and green times (the RMSE is about 1.5 s). There is one missing cycle in this case, as shown in Fig. 7(a). The cycle has three vehicle arrivals, as depicted using triangles. None of them, however, are sampled, i.e., the signal timing estimation algorithms cannot "see" their existence. Using the missing cycle detection method, we can still properly estimate the boundaries of this cycle. Fig. 7(b) shows results for the NGSIM data under 20% penetration. There are in total nine cycles. Five of them have only one sample, which is both CBV and CEV. There is no sample for the second and third cycles. The missing cycle identification algorithm shown in Fig. 5 converges at $m_n = 2$, which correctly detected these two missing cycles. The estimated cycle boundaries also match well with the actual cycle boundaries, as shown in the figure—the RMSE is about 8 s. The RMSE for the NGSIM data is higher since the penetration is lower (20% instead of 50% as for the simulation data).

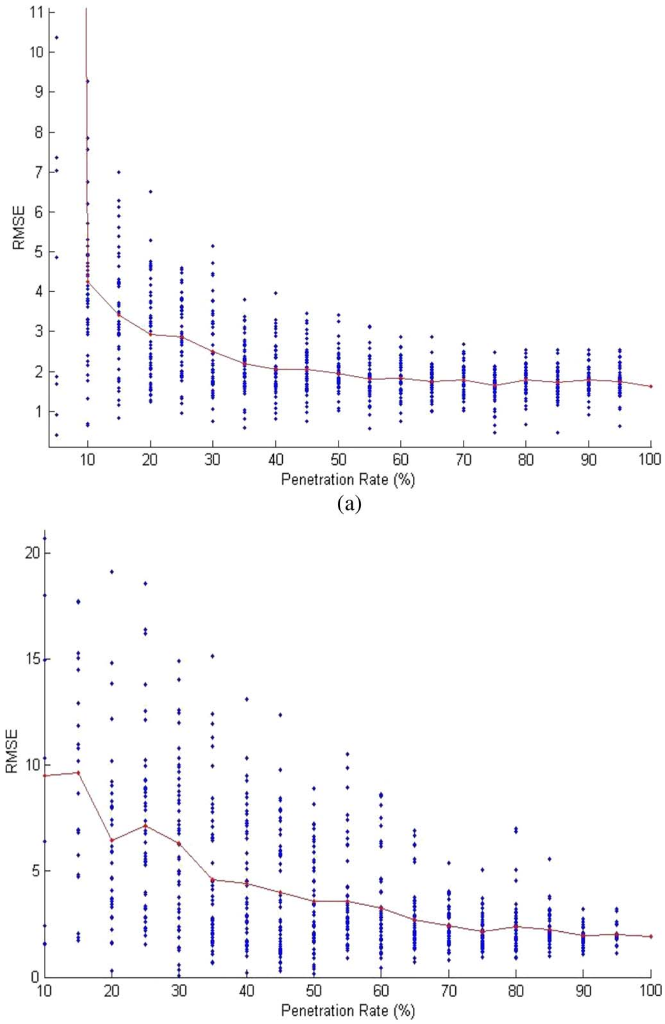


(a)

(b)

Fig. 7. Signal timing estimation. (a) Simulation data (50% penetration rate). (b) NGSIM data (20% penetration rate).

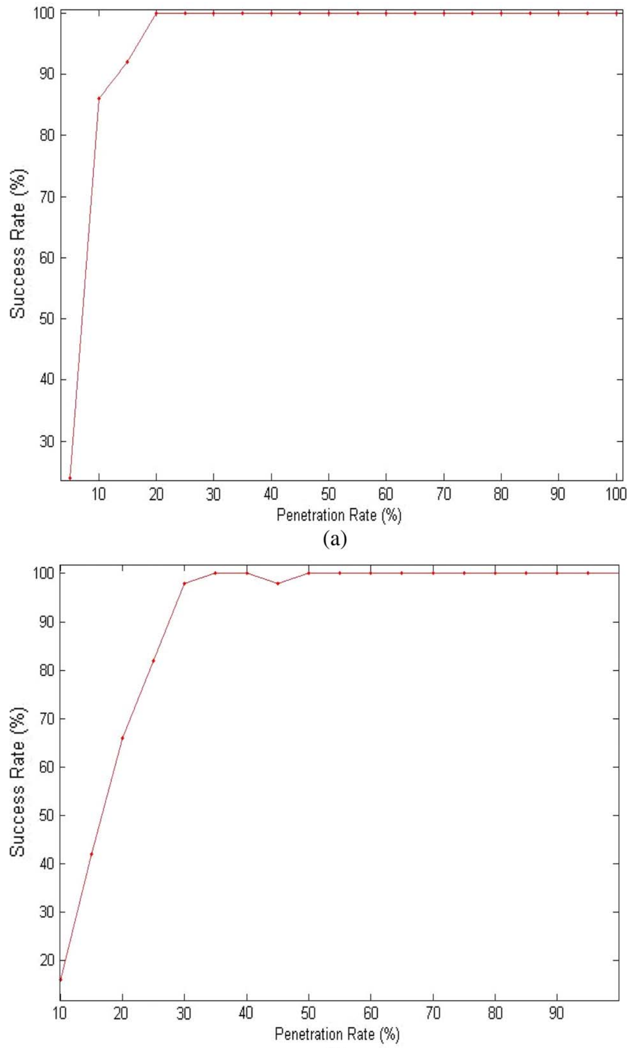
We next show the impacts of the penetration rates on the estimation performance. Fig. 8(a) depicts the RMSEs of the



(a)

(b)

Fig. 8. RMSE versus penetration rate in simulation and NGSIM. (a) Simulation data. (b) NGSIM data.



(a)

(b)

Fig. 9. Success rate versus penetration rate in simulation and NGSIM. (a) Simulation data. (b) NGSIM data.

start of red times for different penetration rates (from 10% to 100% using 5% as the increment and 50 runs for each case) for the simulation. Asterisks for each penetration rate show the RMSEs for all 50 random draws. The solid line depicts the average RMSE for each rate, which shows a roughly decreasing trend as the penetration rate increases. At 5%, the SVM and the cycle boundary detection methods fail to work because the samples are too sparse. When the penetration rate is larger than 10%, relatively accurate estimation results can be achieved, and the RMSE is about 4 s. The error decreases to around 2 s when the penetration rate is larger than 30%. Fig. 8(b) depicts the RMSEs of the start of red times under different penetration rates for the NGSIM data. Similar results can be observed: the RMSEs drop from nearly 10 s to about 5 s when penetration rates increase from 10% to 30% and then decreases to around 3 s when the penetration becomes even larger (e.g., larger than 50%).

C. Performance of Missing Cycle Identification

We performed an experiment to determine how effective the approach is when missing cycles occur. As mentioned in Section III, we say the missing cycle detection algorithm is successful if the minimum missing cycles and maximum missing cycles converge. Otherwise, the algorithm fails. We varied the penetration rate from 10% (5% for simulation data) to 100% and randomly simulate sample vehicle travel times for 50 runs under each rate. The successes rate is defined as the ratio between the number of runs successfully applied by missing cycle detection algorithm and the total number of runs (i.e., 50) for each penetration rate.

Fig. 9(a) depicts the success rates of the missing cycle detection algorithm on simulation data. The figure shows a sudden increase in RMSE at about 10% and below which the algorithm cannot work well. A 100% success rate can be achieved when the penetration rate is greater than 20%.

Fig. 9(b) depicts the success rates of the missing cycle detection algorithm for the NGSIM data. It shows that the success rate is high when the penetration rate is larger than 25%. When the penetration rate is greater than 35%, a 100% success rate can be obtained. The success rate for the NGSIM data looks lower than the simulation data because the NGSIM sample size and the number of cycles (9) are much smaller (the NGSIM testing data were only available for 15 min).

VI. Concluding Remarks

We have presented in this paper a three-step method to estimate the signal timing parameters of an arterial intersection based on sample intersection travel times, including cycle breaking estimation, exact cycle boundary detection, and effective red (green) time estimation. The method was based on the correlation between critical intersection travel time patterns (such as discontinuities) and the signal timing changes (such as the start of the red time). Cycle breaking estimation was cast as an SVM classification problem to identify the first sample vehicle that indicates the start of a cycle (defined as the start of the red time). The cycle breaking algorithm can be applied to any type of signals with or without constant cycle lengths. The exact cycle boundary detection algorithm was formulated as a nonlinear program based on the observation that the exact cycle boundaries must lie between the arrival times of CEV and CBV. The algorithm can only be applied to signals with constant cycle lengths such as pretimed signals or the coordinated phases of actuated signals. The effective red (green) time estimation method was based on the line fitting method in [9] to explore the duration of the effective red time and the time that the delay pattern displays nonsmoothness. The three-step method was tested using data from simulation, a field test, and NGSIM. The results show that accurate cycle breaking and cycle boundary detection results can be achieved under relatively high penetration of travel time data (e.g., $>=$10–15%). Although it can hardly be achieved in the current state of the practice, such high penetration of intersection travel times may be possible in the near future with the deployment of new arterial data collection techniques. The proposed method in this paper can then be used to estimate signal timing parameters, which can be useful to conduct large-scale signal and arterial performance measurement, particularly for those areas where real-time signal information or arterial data collection systems have yet to be established or are not easy to access.

The proposed signal timing estimation approach in this paper, particularly the cycle breaking (SVM-based) and exact cycle boundary detection (optimization-based) methods, is a combination of transportation principles (such as how delay changes are correlated to signal timing) and knowledge (such as signal characteristics: constant cycle or not) with optimization/learning techniques. We believe that methods based on such a combination of traffic principles/knowledge and optimization/learning can play an important role in exploring how to best use travel times and similar forms of data from mobile sensors [4], [8], [9] that have been increasingly available in the field.

For future research, first, it will be of great value if we can extend the current method to deal with all signal types, e.g., the uncoordinated phases of actuated signals or even adaptive signals. For this, the proposed method provides a general framework (such as the three-step process and the idea of combining traffic theories/principles with advanced optimization/learning techniques) that may be extended to capture such signals. For example, the cycle length variable in (8), i.e., C, may be modeled as cycle specific as C_n, which allows the cycle length to vary from cycle to cycle. The authors are currently working on this problem. Second, the method proposed in this paper can directly be applied to estimate the signal timing parameters of a particular movement of an intersection. To obtain a complete picture of the intersection timing setting, one needs to apply the method to all movements. A synthesis procedure is then needed to combine the signal timing information from all separate movements into a coherent phase plan. Such a procedure will be investigated in subsequent papers. Third, the proposed method may be revised to estimate the optimal offset between two adjacent signals. Such information is critical for signal coordination. Forth, extending the proposed method to signal timing design and optimization is an interesting future research topic. Last but not least, the proposed method also needs to be further tested and validated using more data sets collected, particularly from real-world congested signalized intersections. The authors will pursue this direction in future research.

Acknowledgment

The authors would like to thank the anonymous referees for their insightful comments and helpful suggestions on an earlier version of this paper.

References

[1] *Highway Capacity Manual*, Transp. Res. Board (TRB): Washington, DC, 2010.

[2] A. Skabardonis and N. Geroliminis, "Real-time estimation of travel times on signalized arterials," in *Proc. 16th Int. Symp. Transp. Traffic Theory*, 2005, pp. 387–406.

[3] X. Liu, X. Wu, W. Ma, and H. Hu, "Real time queue length estimation for congested signalized intersections," *Transp. Res. Part C, Emerging Technol.*, vol. 17, no. 4, pp. 412–427, 2009.

[4] X. Ban, P. Hao, and Z. Sun, "Real time queue length estimation for signalized intersections using sampled travel times," *Transp. Res. Part C, Emerging Technol.*, vol. 19, no. 6, pp. 1133–1156, Dec. 2011.

[5] J. S. Wasson, J. R. Sturdevant, and D. M. Bullock, "Real-time travel time estimates using media access control address matching," *ITE J.*, vol. 78, no. 6, pp. 20–23, 2008.

[6] K. Kwong, R. Kavaler, R. Rajagopal, and P. Varaiya, "Arterial travel time estimation based on vehicle re-identification using wireless magnetic sensors," *Transp. Res. Part C, Emerging Technol.*, vol. 17, no. 6, pp. 586–606, Dec. 2009.

[7] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J. Herrera, and A. Bayen, "Virtual trip lines for distributed privacy-preserving traffic monitoring," in *Proc. Int. Conf. Mobile Syst., Appl., Services*, 2008, pp. 15–28.

[8] J. C. Herrera, D. B. Work, R. Herring, X. Ban, and A. Bayen, "Evaluation of traffic data obtained via GPS-enabled mobile phones: The mobile century field experiment," *Transp. Res. Part C, Emerging Technol.*, vol. 18, no. 4, pp. 568–583, Aug. 2009.

[9] X. Ban, R. Herring, P. Hao, and A. Bayen, "Delay pattern estimation for signalized intersections using sampled travel times," *Transp. Res. Rec.*, vol. 2130, pp. 109–119, 2009.

[10] Google, *The Bright Side of Sitting in Traffic: Crowdsourcing Road Congestion Data*, Jul. 20, 2011. [Online]. Available: http://googleblog.blogspot.com/2009/08/bright-side-of-sitting-in-traffic.html

[11] Inrix, *INRIX Traffic*, Jul. 20, 2011. [Online]. Available: http://www.inrixtraffic.com/

[12] B. E. Boser, I. M. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, Pittsburgh, PA, 1992, pp. 144–152.

[13] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, Jun. 1998.

[14] O. L. Mangasarian, "Generalized support vector machines," in *Advances in Large Margin Classifiers*, A.J. Smola, P. Bartlett, B. Schokopf, and D. Chuurmans, Eds. Cambridge, MA: MIT Press, 2000, pp. 135–146.

[15] K. P. Bennett and C. Campbell, "Support vector machines: Hype or hallelujah?" *SIGKDD Explorations Newslett.*, vol. 2, no. 2, pp. 1–13, 2000.

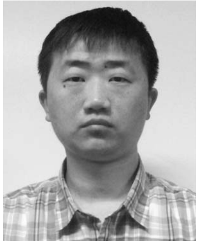
[16] C. Cortes and V. Vapnik, "Support vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[17] P. Hao and X. Ban, "Vehicle queue location estimation for signalized intersections using sample travel times from mobile sensors," submitted for publication.

[18] H. Liu and S. Jabari, "Evaluation of corridor traffic management and planning strategies using microsimulation: A case study," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2088, pp. 26–35, 2008.

[19] "Final report of task order 3: Corridor management plan demonstration," California Center Innov. Transp., Univ. California, Berkeley, CA, 2006.

[20] Cambridge Systematics, *Summary Rep.: NGSIM Peachtree Street (Atlanta) Data Analysis (4:00 p.m. to 4:15 p.m.)*, 2006. Accessed on May 8, 2011. [Online]. Available: http://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm



Peng Hao received the B.S. degree in civil engineering from Tsinghua University, Beijing, China, in 2009. He is currently working toward the Ph.D. degree with the Department of Civil and Environmental Engineering, Rensselaer Polytechnic Institute, Troy, NY.

His research interests are intelligent transportation systems and sensor-aided modeling and simulation. His current research focuses on application of traffic flow theory and machine learning techniques (such as support vector machines and Bayesian networks) to transportation modeling.

Xuegang (Jeff) Ban received the M.S. degree in computer sciences and the Ph.D. degree in transportation engineering from the University of Wisconsin, Madison.

He spent three years as a Post-Doctoral Researcher with the Institute of Transportation Studies, University of California, Berkeley. He is currently an Assistant Professor with the Civil and Environmental Engineering Department, Rensselaer Polytechnic Institute, Troy, NY. His research interests are in transportation network modeling, traffic modeling and simulation, and intelligent transportation systems. His current research focuses on how to use mobile traffic sensors (such as Global Positioning Systems (GPS) enabled cellular phones) as traffic probes to evaluate real-time traffic system performances and how to systematically integrate those sensors into dynamic traffic network analysis and management. He has published over 50 papers in peer-reviewed journals and conferences. His research has been supported by the National Science Foundation, U.S. Department of Transportation (USDOT), Caltrans, New York State Energy Research and Development Authority, and the New York State Department of Transportation. He now serves on the editorial boards of the *Journal of Intelligent Transportation Systems* and *Networks and Spatial Economics*.

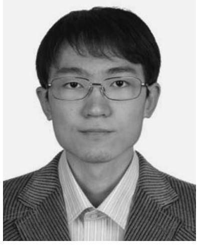
Kristin P. Bennett received the B.S. degree in mathematics and computer science from the University of Puget Sound, Tacoma, WA, and the M.S. and Ph.D. degrees in computer sciences from the University of Wisconsin, Madison.

She is currently a Professor with the Mathematical Sciences and Computer Science Departments, Rensselaer Polytechnic Institute, Troy, NY. She is an active member of the machine learning, data mining, and operations research communities, serving as present or past Associate or Guest Editor for ACM *Transactions on Knowledge Discovery from Data*, the SIAM *Journal on Optimization*, *Naval Research Logistics*, the *Machine Learning Journal*, the IEEE TRANSACTIONS ON NEURAL NETWORKS, and the *Journal on Machine Learning Research*. She served as Program Chair of the 11th ACM Special Interest Group on Knowledge Discovery and Data Mining International Conference on Knowledge Discovery and Data Mining. She has been researching mathematical-programming approaches to machine learning such as support vector machines since 1989. In addition, she has worked extensively on the successful application of machine learning to problems in chemistry, biology, epidemiology, engineering, and business.

Qiang Ji (SM'04) received the Ph.D. degree in electrical engineering from the University of Washington, Seattle.

He recently served as a Program Director with the National Science Foundation (NSF), where he managed NSF's computer vision and machine learning programs. He also held teaching and research positions with the Beckman Institute, University of Illinois at Urbana-Champaign; the Robotics Institute, Carnegie Mellon University; the Department of Computer Science, University of Nevada at Reno; and the U.S. Air Force Research Laboratory. He is currently a Professor with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute Troy, NY. He is currently serves as the Director of the RPI Intelligent Systems Laboratory. He has published over 150 papers in peer-reviewed journals and conferences. His research has been supported by major governmental agencies, including NSF, the National Institutes of Health, the Defense Advanced Research Projects Agency, the Office of Naval Research, the Army Research Office, and the Air Force Office of Scientific Research, as well as by major companies including Honda and Boeing. His research interests are in computer vision, probabilistic graphical models, information fusion, and their applications in various fields.

Dr. Ji is an Editor for several related IEEE and international journals and has served as Chair, Technical Area Chair, and Program Committee at numerous international conferences/workshops.



Zhanbo Sun received the B.S. degree in civil engineering from Tsinghua University, Beijing, China, in 2009. He is currently working toward the Ph.D. degree with the Department of Civil and Environmental Engineering, Rensselaer Polytechnic Institute, Troy, NY.

His research interests are intelligent transportation systems and traffic modeling and simulation. His current research focuses on the modeling and privacy protection aspects of using mobile sensors as traffic probes.