

Delay Pattern Estimation for Signalized Intersections Using Sampled Travel Times

Xuegang (Jeff) Ban, Ryan Herring, Peng Hao, and Alexandre M. Bayen

Intersection delays are the major contributing factor to arterial delays. Methods to estimate intersection delay patterns by using measured travel times are studied. The delay patterns provide a way to estimate the delay for any vehicle arriving at the intersection at any time, which is useful for providing time-dependent intersection delay information to the driving public. The model requires sampled travel times between two consecutive locations on arterial streets, one upstream and the other downstream of a signalized intersection, without the need to know signal timing or traffic flow information. Signal phases can actually be estimated from the delay patterns, which is a unique feature of the proposed method in this paper. The proposed model is based on two observations regarding delays for signalized intersections: (a) delay can be approximately represented by piecewise linear curves due to the characteristics of queue forming and discharging and (b) there is a nontrivial increase in delay after the start of the red time that enables detection of the start of a cycle. A least-squares-based algorithm is developed to match measured delays in each cycle by using piecewise linear curves. The proposed model and algorithm are tested by using field experiment data with reasonable results.

Travel time or delay is one of the most important roadway traffic metrics. Providing travel times on freeway routes, for example, via freeway changeable message signs, has now become a common practice in many states in the United States. Arterial travel time information, however, is not widely available due to the difficulty of estimating arterial traffic conditions. Arterial traffic is fundamentally different from freeway traffic. The difference in traffic flow patterns is mainly due to the existence of traffic signals, stop signs, and cross traffic that introduces interruptions to arterial traffic flow. These interruptions bring discontinuities to quantities of interest such as travel times or delays. In addition, distinct from freeways, in an arterial network there are usually many possible routes from an origin to a destination. Providing travel times for one or a few routes may not be sufficient for a driver to get a full picture of the arterial traffic conditions. Therefore, providing time-dependent delay information for arterial

intersections seems more desirable. This paper focuses on signalized intersection delays as they are typically the major contributing factor to arterial delays.

Various models focusing on signalized intersections have been developed to estimate arterial travel times or delays. Statistical methods are proposed in (1–3) in which travel times are modeled as a linear combination of occupancy, flow, and signal parameters. Xie et al. (4) treat arterial link travel time as the summation of cruise time and signal delay. Cruise time is computed by using detector speeds, and signal delay is estimated by using a simplified intersection queuing diagram that requires basic signal parameters. Skabardonis and Dowling developed an improved speed–flow relationship (5) that was shown to be effective to calculate arterial link travel times (6). These models are mainly for estimating average (or static) arterial travel times; recent attention has focused on estimating dynamic (or time-dependent) arterial travel times (7, 8). Skabardonis and Geroliminis (7) model link travel time as the summation of free-flow travel time and signal delay; signal delay consists of single vehicle delay, queuing delay, and oversaturation delay. The calculation of signal delay requires 30-s traffic volume and detailed signal timing parameters. By using high-resolution (second-by-second) traffic signal events data (such as phase and timing changes) and vehicle actuation data, Liu and Ma (8) construct “virtual” vehicle trajectories that make it possible to estimate accurate dynamic arterial travel times.

Most existing arterial models require, as a minimum, the knowledge of traffic signal timing parameters and traffic volume to estimate arterial travel times or delays. Collecting traffic signal data for wide-area arterial streets is not trivial since historically traffic signals have been operated and maintained by multiple agencies. By using the vehicle reidentification technique, it has been shown (9–11) that samples of intersection delays can be obtained directly. In particular, Kwong et al. (12) propose a new scheme in which wireless traffic sensors are deployed downstream (at a fixed distance such as 12 m) of signalized intersections. Traffic volume is collected at each sensor location together with vehicle signatures. A specially designed vehicle reidentification algorithm is developed to match vehicles from signatures (12). The algorithm is based on a statistical model of the signatures, with parameters estimated from data, and no “ground truth” is required. If the algorithm is applied to two consecutive sensor locations (one upstream and the other downstream of a signalized intersection), intersection travel times (or delays) can be obtained directly. A unique feature of such a vehicle reidentification method is that traffic signal information is not required. It is further shown that signal phases can be derived from the matched vehicles by looking at the start and end times of the first vehicle in a queue (12).

The vehicle reidentification method provides a straightforward way for estimating intersection delays without the requirement of

X. Ban and P. Hao, Department of Civil and Environmental Engineering, Rensselaer Polytechnic Institute, Room JEC 4034, 110 Eighth Street, Troy, NY 12180-3590. R. Herring, Department of Industrial Engineering and Operations Research, University of California, Berkeley, 2105 Bancroft Way, Suite 300, Berkeley, CA 194704. A. M. Bayen, Department of Civil and Environmental Engineering, Systems Engineering, University of California, Berkeley, 711 Davis Hall, Berkeley, CA 94720-1720. Corresponding author: X. Ban, banx@rpi.edu.

Transportation Research Record: Journal of the Transportation Research Board, No. 2130, Transportation Research Board of the National Academies, Washington, D.C., 2009, pp. 109–119.
DOI: 10.3141/2130-14

signal information. Sampled travel times, however, only provide discrete measurements in the time domain. Now the question is: Can a time-dependent intersection delay pattern curve be constructed by using sampled travel times for a given signalized intersection? An intuitive answer is to assume travel times change linearly between two neighboring sampled travel times. As shown below, such a method may not be the most effective, especially when the penetration rate is relatively high. In this paper, a least-squares-based algorithm is developed to estimate the delay patterns from sampled travel times by recognizing the underlying characteristics of signalized intersection delays.

The proposed algorithm can be applied to specially deployed fixed-location sensors [such as loop detectors or wireless sensors (12)] or the virtual trip line (VTL) technique based on global positioning system (GPS)-equipped cell phones (13, 14). VTLs are virtual loop detectors without any requirement to deploy physical detectors or other infrastructures. As a vehicle equipped with a GPS cell phone passes by a VTL location, the location and speed of the vehicle are sent to a secure server from which all vehicles' information is aggregated and transferred to traffic models. Deployment of VTLs is flexible, with major considerations for privacy preservation (15). Arterial VTL data include individual vehicle speeds at each VTL and travel times between consecutive VTLs for vehicles equipped with GPS cell phones. Such data provide rich information about arterial traffic states while maintaining privacy violations at a minimal level.

The raw VTL travel time data can be processed to generate samples of intersection delays. In this paper, methods are proposed to estimate intersection delay patterns by using these samples. The authors show that delay patterns can be represented as piecewise linear (PWL) curves. These curves are developed by using well-developed traffic flow theory on queue forming and discharging at signalized intersections. The authors then show how to use collected VTL travel times to estimate the parameters of pattern curves, without knowing either traffic signal parameters or traffic volume. The estimation algorithm is a two-step least-squares method that can be converted to solve multiple convex quadratic programs in small dimensions. The estimated delay patterns can also be directly used to derive signal phases. The model and algorithm are tested in microscopic traffic simulation and validated by using field experiment data obtained from wireless sensors.

ARTERIAL VTL SYSTEM

For arterials, VTLs are deployed in a similar way as wireless traffic sensors are deployed (12). In general, a VTL is placed downstream of each outgoing approach of an intersection. The type of data generated by the VTL system for a pair of VTLs includes the time crossing the first VTL, travel time between the two VTLs, and the average speed when vehicles cross each VTL. Speeds are unlikely to be useful because they are highly variable around intersections. Instead, the travel time information will be used to measure delays through the intersection. Given that there will be VTLs deployed to all sides of an intersection, travel time information will be obtainable for any turns of the intersection in addition to through traffic. The fully deployed VTL system will collect updates and push them to a server for processing. The time between pushes will be a consistent interval, typically 1 min.

APPROXIMATE INTERSECTION DELAY PATTERNS

PWL Intersection Delay Curves

Models are first derived for approximate patterns of intersection delays under normal and oversaturation conditions. The results presented here are based on well-established theories on queue forming and discharging in front of a signalized intersection (16, 17). The first condition occurs when the queue can be cleared completely during the green phase of a cycle; the second condition refers to situations in which the queue cannot be cleared within one cycle and the residual queue must wait for extra time (i.e., more delays) to be cleared. These two conditions are the most commonly observed in the field. Under specific situations (e.g., heavy congestion), queues may spillover to upstream intersections and cause further delays. This third condition is not considered in this paper and will be studied in future research.

Figure 1a depicts a typical signalized intersection with VTLs installed upstream (VTL1) and downstream (VTL2). To simplify the discussion, assume that the queue never passes VTL1. The bold solid triangles in the figure can be used to represent how the queue forms and dissipates (these triangles show the waves where two distinct traffic states meet). The horizontal part of the triangles represents the duration of red time. If delays due to vehicle decelerations and accelerations are ignored and the arrival rate is uniform within one cycle, delays can be fully determined by the triangles. In the figure, dashed lines represent trajectories of vehicles, while dotted lines are boundaries at which the discontinuities of delays occur.

The authors' aim is to characterize vehicle delays as a function of the time when a vehicle passes VTL1. In reality the measured delay will not be recognized until the vehicle passes VTL2, but here it is assumed that data have been collected and thus one can perform post-processing to reconstruct a mapping from the time that a vehicle passed VTL1 to its experienced delay at the intersection. Since it is assumed that the queue never reaches VTL1, as shown by the trajectories of vehicles (dashed lines), if a vehicle approaches the intersection in red time or if the queue length is not zero (e.g., trajectory *a* in the figure), then the vehicle will join the end of the queue first and thus be delayed. The delay encountered by the vehicle is the horizontal part of trajectory *a*. Otherwise, if a vehicle arrives during green time and there is no queue (e.g., trajectory *b*), the vehicle will pass the intersection with no delay. The (red) delay curve at the bottom of Figure 1a will spike up at the time that allows a vehicle to travel to the intersection in free flow just before the start of the red time. More importantly, by analyzing the geometry of the triangles, one can observe that if a vehicle passes by VTL1 at a time that would make it get to the intersection just after the start of the red time, delay for this vehicle will be the maximum for the specific cycle. After that, delays will be reduced linearly until no delay is reached. This is represented by the line segments marked as "1" of the delay curve at the bottom of Figure 1a. The slope of the delay reduction part, denoted as delay reduction rate *s*, can be calculated analytically as

$$s = \frac{u_f(w - u_w)}{w(u_f + u_w)} = \frac{v}{k_j} \left(\frac{1}{u_f} + \frac{1}{w} \right) - 1 \quad (1)$$

where

- w = wave speed,
- u_f = free-flow speed,
- u_w = wave speed when a vehicle joins the queue,
- k_j = jam density, and
- v = traffic flow.

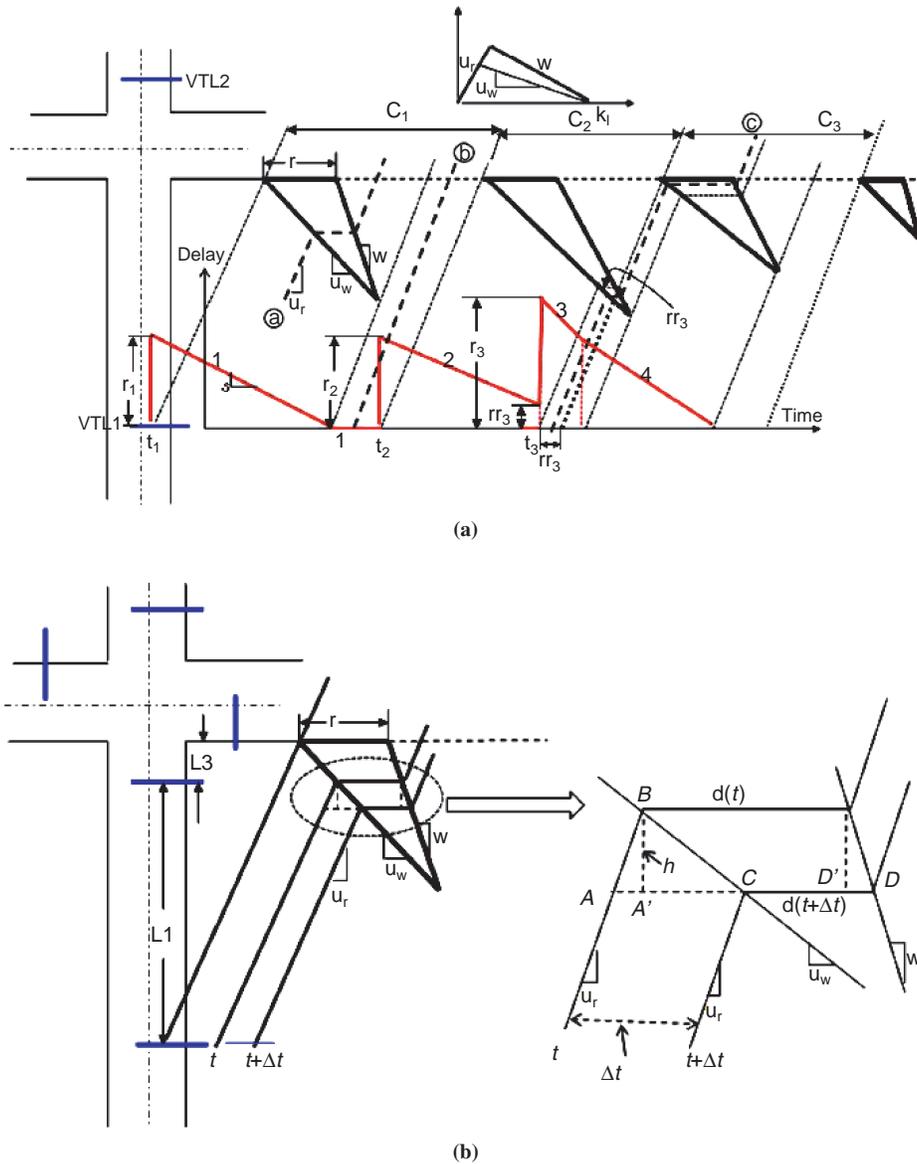


FIGURE 1 Theoretical delay patterns: (a) intersection delay patterns and (b) calculation of delay reduction rate.

Traffic flow (v) is assumed to be constant within a cycle. The parameters u_f , w , and k_j are specific to actual arterial locations, which also determine the fundamental diagram of the location. Since $w \geq u_w$ always holds (refer to the fundamental diagram at the top of Figure 1a), s is nonnegative: the delay always reduces from its maximum (when traffic light turns red) to some minimum value (when light turns green and no queue exists) for normal situations.

To illustrate how Equation 1 can be derived, see Figure 1b. In particular, it is assumed the delays for a vehicle passing VTL1 at time t and $t + \Delta t$ are $d(t)$ and $d(t + \Delta t)$, respectively. According to the assumptions made in this paper, the delays at both time instants correspond to the lengths of the horizontal lines as shown in Figure 1. On the basis of the geometry of the triangles,

$$d(t + \Delta t) - d(t) = \overline{CD} - \overline{A'D'} = \overline{D'D} - \overline{A'C} = \frac{h}{w} - \frac{h}{u_w}$$

and

$$\Delta t = \frac{h}{u_f} + \frac{h}{u_w}$$

and therefore

$$h = \frac{\Delta t}{\frac{1}{u_f} + \frac{1}{u_w}}$$

Equation 2 summarizes these equations:

$$d(t + \Delta t) - d(t) = \Delta t \left(\frac{1}{w} - \frac{1}{u_w} \right) \left(\frac{1}{\frac{1}{u_f} + \frac{1}{u_w}} \right) \tag{2}$$

Since the delay reduction rate can be defined as

$$s = \frac{d(t + \Delta t) - d(t)}{\Delta t}$$

Equation 1 can be obtained via dividing both sides of Equation 2 by Δt .

The above analysis and Equation 1 work only for normal conditions, that is, no oversaturation or spillover occurs. In case of oversaturation, the residual queue from one cycle will have to wait for the next green to be cleared, as shown by trajectory *c* in Figure 1*a*. Under such situations, delay will still be reduced linearly from the maximum value after the start of the red time. However, it will never reach zero; instead, it will have a sudden increase from a nonzero delay to another (local) maximum, indicating the vehicle will have to wait for extra cycle(s) to be cleared. This is marked as “2” in the delay curve in Figure 1*a*. After this stage, the delay will be reduced linearly until the impact of the residual queue diminishes, as shown by “3” in the delay curve. The delay will be further reduced in a normal way as marked by “4” in the curve. As a result, the delay curve for oversaturation is still PWL, but with a more complicated pattern. A distinct feature is that delay is never reduced to zero. The slope of the curves can all be computed analytically by looking at the geometry of the triangles in Figure 1*b*. It can be seen that the approximate delay patterns for signalized intersections (by ignoring the acceleration and deceleration delays) can be represented as PWL curves. The curves are continuous in most cases, but contain discontinuities (jumps) periodically. These discontinuities correspond to the start of red times and are important features of intersection delays.

Estimation of Signal Phases from PWL Intersection Delay Pattern

Knowing the PWL intersection delay pattern enables one to estimate signal phases of the intersection. Here it is assumed a cycle always starts with the red time, implying that the start of the red, the duration of the red, and the end time of a cycle (also the start time of the next red) uniquely determine the cycle. Figure 1*a* shows a focus on the translated signal phase timing (TSPT) at VTL1, which is different from the actual signal phase timing at the intersection by a constant (i.e., the free-flow travel time from VTL1 to the intersection). In fact, TSPT reflects the times when a vehicle actually “feels” the effect of the signal at VTL1 as if it were just at the intersection. The procedure for estimating TSPT is described as follows.

First, as shown in Figure 1*a*, there is a nontrivial increase in delays right after the start of the red time in TSPT (for both normal and oversaturation conditions), with the magnitude of the increase equal to the duration of red time. As delay generally decreases over time within a cycle after the start of red, such an increase is a unique feature of intersection delays that only happens at the time when the signal turns red in TSPT. Detecting such an increase in measured delays will help to identify the start of a new cycle. For example, under normal conditions (e.g., the condition marked as “1” of the delay pattern in Figure 1*a*), the delay increases from 0 to r_1 at t_1 , which indicates that t_1 is the start of a cycle (denoted as cycle “ C_1 ”) in TSPT. This cycle ends when the next increase is detected at time t_2 , which also indicates that the next cycle (C_2) starts at t_2 . The duration of the red time is r_1 . For oversaturation conditions, the start of red is also associated with such an increase in delay, but needs further adjustment. For example, at t_3 the delay increases from a nonzero value rr_3 to r_3 . As illustrated in Figure 1, the actual start of red (for cycle C_3) in this case is not t_3 ; rather, it is $t_3 + rr_3$. Similarly, the duration of red is $r_3 - rr_3$ instead of r_3 .

In summary, it is assumed the delay pattern is given, which results in n discontinuities at t_i with delay being increased from rr_i to r_i , $i = 1, \dots, n$. The start of red time is then $t_i + rr_i$, and the duration of red is $r_i - rr_i$, $i = 1, \dots, n$. This simple procedure is used in later sections to derive signal phase information for both the simulation and field experiment data. Notice that this way phase information in TSPT is obtained, which can be easily translated to actual phase information of the intersection by adding the free-flow travel time from VTL1 to the intersection.

ESTIMATION ALGORITHM

The problem investigated in this article is to estimate intersection delay patterns by using sampled travel times measured between upstream and downstream locations of a signalized intersection. The estimation method proposed is a simple curve fitting algorithm. First, since delay curves are PWL, delay measurements can be fitted by using linear forms, which significantly reduces the complexity of the fitting algorithm. Second, there is a nontrivial increase in delays right after the start of the red time; detecting such an increase can help identify the start of a new cycle.

Two-Step Least-Squares Estimation Algorithm

The estimation algorithm contains cycle breaking and line fitting as two major steps. Figure 2 shows how the above two observations

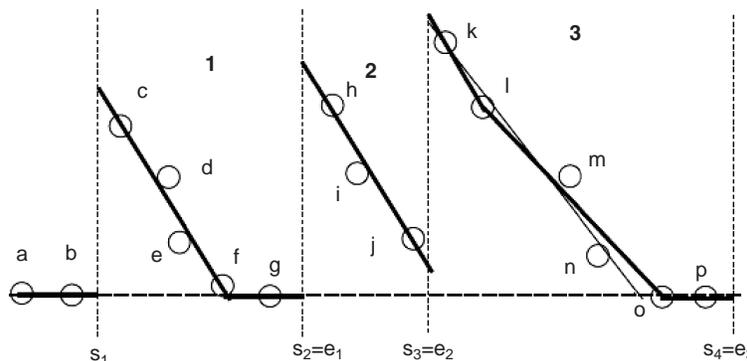


FIGURE 2 Illustration of estimation algorithm.

can be used in these steps. It is assumed there are 16 measured delays, represented by the 16 circles labeled “a” to “p” in Figure 2. The values of the delays are denoted as $\{d_r, 1 \leq r \leq 16\}$. Each delay is associated with a time stamp, denoted as $\{t_r, 1 \leq r \leq 16\}$. First, by detecting the (nontrivial) increase of delays, the 16 measurements can be broken into four groups: $\{a, b\}$, $\{c, d, e, f, g\}$, $\{h, i, j\}$, $\{k, l, m, n, o, p\}$. In the figure, s_i, e_i denotes the starting time and ending time of a cycle, respectively, which may be defined as the middle point of two consecutive time stamps (one in each cycle) or adjustable on the basis of the calculated average cycle length information (refer to the intersection delay estimation [IDE] algorithm in the next subsection). Second, within each cycle, an attempt is made to fit the measurements by using PWL curves. Figure 2 shows the three typical delay patterns for normal and oversaturation conditions, marked as 1, 2, and 3. Curve 1 is for normal conditions, in which delay reduces linearly until it reaches zero. Therefore, the delay curve consists of two lines, one with a negative slope and the other with a zero slope (constant). Curves 2 and 3 are for oversaturation conditions. Curve 2 is a single line with a negative slope representing delay reduction (over time) for the first cycle of the oversaturation, in which the minimum delay is positive. Curve 3 represents the delay reduction pattern caused by both cycles of the oversaturation, which contains at least two lines and may or may not reach zero delay in the end (depending on whether oversaturation disappears in the second cycle).

In summary, although the shape of the delay curve within a cycle may vary depending on actual traffic conditions, the fundamental pattern of the curve can be identified as the three cases in Figure 2 for normal and oversaturation conditions. Furthermore, the number of measurements in one cycle tends to be small. For example, consider a three-lane arterial intersection with a total traffic volume of 1,800 vehicles per hour (veh/h). Assume the cycle length is 1 min, which will result in about 30 (1,800/60) measurements under a 100% penetration rate. In reality, since the penetration rate is most likely much smaller than 100%, the number of measurements with a cycle will not exceed one or two dozen. Therefore, although more advanced fitting techniques may be applied, instead a simplistic method based on least-squares fitting is proposed in this paper.

The least-squares method starts with attempting to fit the measurements within one cycle by using two straight lines. This is done by enumerating all possible grouping scenarios of the measurements. Denote $\{d_r, r \in R\}$ the set of measurements sorted by their time stamps $\{t_r, r \in R\}$, where $|R|$ denotes the total number of measurements. This set of measurements may then be divided into two groups by breaking the set at $m = 3, \dots, |R| - 1$, where m is the starting index of the second group. For each m , fitting can be solved by using a convex quadratic program. To see this, it is assumed the objective of fitting is to reduce the deviation of model-predicted and actually measured delays, more specifically the mean square error (MSE) of the predicted delays. It is further assumed the first line can be represented as $d = a_1 t + b_1$ and the second line as $d = a_2 t + b_2$. Here a_1, b_1 are parameters for the first line, and a_2, b_2 are parameters for the second line; all need to be estimated. The quadratic problem can then be formulated as follows:

$$\min_{a_1, b_1, a_2, b_2} \sum_{1 \leq i \leq m-1} (a_1 t_i + b_1 - d_i)^2 + \sum_{m \leq i \leq |R|} (a_2 t_i + b_2 - d_i)^2 \quad (3)$$

such that

$$a_1 [(1-\theta)t_{m-1} + \theta t_m] + b_1 = a_2 [(1-\theta)t_{m-1} + \theta t_m] + b_2 \quad (4)$$

In the above model, the objective in Equation 3 is the summation of MSE of the two groups. The first group contains data points $1, \dots, m-1$, and the second group contains data points $m, \dots, |R|$. It is assumed m is given, and $a_1 t_i + b_1$ is the predicted delay at t_i using the first line whose actual delay is d_i for any $1 \leq i \leq m-1$. Similarly, $a_2 t_i + b_2$ is for the delay predicted by the second line. The Equation 4 constraint is required because the two lines have to intersect at the boundary of the two groups. Here it is assumed the boundary is at $0 \leq \theta \leq 1$ from t_{m-1} with respect to the difference between t_m and t_{m-1} . As a special case, if the boundary is at the middle point of t_m and t_{m-1} , then $(t_{m-1} + t_m)/2$. The above model has only four variables and is convex and quadratic, which can be solved very efficiently by using standard quadratic program solvers.

The quadratic model (Equations 3 and 4) will be solved for any $3 \leq m \leq |R| - 1$, resulting in $|R| - 3$ solves. The minimum objective value of all solves is denoted as f_2 . The value for f_2 is compared with the objective value of fitting all measurements by using one line, denoted as f_1 . If $f_2 < f_1$, the two-line fitting is accepted; otherwise, the one-line fitting is accepted. If two-line fitting is accepted, the algorithm will further test if the duration of either group is larger than a threshold. If yes, the above process is repeated on the group, trying to fit the group with two new lines. This process repeats itself until either all groups are represented as a single line or the duration of the group is below the threshold. The estimation algorithm is summarized as follows, which is denoted as the IDE algorithm.

IDE Algorithm

Step 1. Initialization. Collect VTL travel time data and process them to obtain intersection delays. Set two thresholds, th_1 and th_2 .

Step 2. Cycle breaking. Scan all the delay measurements and detect if the delay increase from one measurement to the next one exceeds th_1 . If yes, break the cycle at the second measurement. This step will produce groups of delay measurements.

Step 3. Curve fitting within a cycle. Denote $\{d_r\}, \{t_r\}, \forall r \in R$ all the measurements in a given cycle.

Step 3.1. Solve the convex quadratic program (Equations 3 and 4) for all $3 \leq m \leq |R| - 1$. Here $\theta = 0.5$ is used, that is, the boundary is at the middle point. Denote the minimum objective value among all $|R| - 3$ solves as f_2 .

Step 3.2. Solve the least-squares fitting problem by using a single line and denote its objective value as f_1 .

Step 3.3. If $f_2 > f_1$, fit the delay pattern by using the single line. Otherwise, represent the delay curve by using two lines. If the duration of either line is larger than th_2 , set $\{d_r\}, \{t_r\}, \forall r \in R$ as the measurements corresponding to this line and go to Step 3.1. Go to Step 4 if Step 3 is done for all cycles.

Step 4. Cycle length adjustment (for pretimed or actuated coordinated signals). Calculate the average cycle length by dividing the total time period by the number of cycles detected. Using this average cycle length, adjust the boundaries of each cycle (i.e., the values of θ) so that each cycle length is as close as possible to the obtained average cycle length.

Step 5. Stop with an optimized delay pattern curve.

In the IDE algorithm, th_1 is the threshold for the increase of delays to detect the start of a new cycle, while th_2 is the threshold of the time window to break measurements within a group into possibly more cycles. The value of th_1 should be exactly the duration of the red time in ideal situations. In reality, due to travel time variations

across individual vehicles and more importantly the fact that only samples are available, one can set $th_1 = \alpha_1 R$, where R is the duration of the red time and α_1 is a coefficient. Similarly, th_2 can be selected as the cycle length in ideal cases. In practice, one can set $th_2 = \alpha_2 R$, where C is the cycle length and α_2 is another coefficient. The selections of α_1 and α_2 may be location specific and need further investigations. In this paper, th_1 and th_2 are set as 15 and 35 s, respectively. Step 4 is a fine-tuning step for pretimed or actuated coordinated signals. For these types of signals, cycle lengths are usually constants. The average cycle length via the first three steps can hopefully provide an indication of what the fixed cycle length might be. This information can then be used to adjust boundaries (i.e., θ) of each cycle so that the cycle length is close to the average length.

The above discussions show that in order to appropriately estimate the delay curves at least two measurements per cycle are needed for normal conditions. For oversaturation conditions, this number will be at least four. If the cycle length is 1 min, the required minimum sample rate is 120 veh/h for normal conditions and 240 veh/h for oversaturation conditions. If a two-lane arterial street with a traffic volume of 1,200 veh/h is considered, this implies a minimum penetration rate of 10% for normal conditions and 20% for oversaturation conditions. It is worth noting that the IDE algorithm only uses measured travel times as input, without assuming knowledge of signal timing parameters or traffic volume information. This is a fundamental difference between IDE and previous models based on detector data.

Test of the Algorithm in Microsimulation

The performance of the IDE algorithm is assessed by using a simulation model developed in Paramics (18). The left-turn movement of a particular intersection with a free-flow travel time of 26.69 s is considered. The simulation was run for 1 h. Figure 3a depicts the simulated travel times between two VTLs deployed upstream and downstream of the intersection for all vehicles making left turns. The travel times look purely random at first glance. For comparison purposes, the durations of red times for this left turn are displayed as the horizontal bars at the bottom of Figure 3a. These durations are “ground-truth” and are obtained directly from the simulation model. The IDE algorithm was applied on the simulated travel times; the identified delay patterns are shown in Figure 3b. In this figure, the curves are actually for travel time patterns, which is exactly the same as the delay patterns (with a constant difference). The estimated patterns match very well with the measured travel times (represented as asterisks in Figure 3); the estimation errors, that is, $\hat{d}_i - d_i$, are indicated by plus signs. Here \hat{d}_i is the estimated delay. It is easy to see that most estimation errors are close to zero, implying that the estimation quality is high. To further quantify the estimation quality, a quality measure is defined that is the percentage of estimates with errors no more than 15% of the measured travel times. Denote this quality measure as α , which can be defined as follows:

$$\alpha = \text{Prob} \left(\left| \frac{\hat{d}_i - d_i}{d_i + \text{fft}} \right| \leq 0.15 \right) \quad (5)$$

Here fft denotes the free-flow travel time. Note that fft is added to the denominator of the right side of Equation 5 since d_i may be zero. In this sense, Equation 5 is actually the error defined for travel times. Clearly, the estimation quality becomes higher for larger α . In this example, $\alpha = 99.32\%$, which indicates that the IDE algorithm works well for estimating delay patterns. Notice that during this

1-h simulation, both normal conditions and oversaturation conditions occur (Figure 3b). These conditions are verified in the actual simulations. Also, by comparing the delay patterns with the ground-truth red times on the bottom of the figure, it can be further verified the patterns are associated with signal timing properly.

The good performance of the IDE algorithm in the above example is largely due to the fact that all vehicle travel times are assumed to be known. In other words, the penetration rate is 100%. The next question to ask is: How will penetration influence the estimation quality? To answer this question, the measured travel times for a given penetration rate p are randomly sampled, and the sampled travel times are used to estimate delay patterns. For this purpose, it is assumed the probability of selecting a particular measurement is p . The sampling results in two sets: the first set contains travel times that were selected, and the second set consists of all unselected travel times. The first set is used to estimate delay patterns via the IDE algorithm; the second set is used for testing the estimation quality.

Figure 4a depicts the impacts of penetration rates on the estimation quality; the penetration rate is varied from 6% to 100% using 2% as the increment. For each penetration rate, the random sampling procedure was run 50 times. Each time, the sampled travel times were used to estimate delay patterns, and the unselected travel times were used to test the estimation quality, that is, to compute α . The plus signs in Figure 4a represent the α 's and the solid line is the average of the 50 runs. For comparison purposes, the estimation was also calculated by pure linear interpolation. That is, for each sampling run, the sampled travel times are treated as grid. The unselected travel times can then be estimated by assuming travel times change linearly between any two adjacent travel times. This linear interpolation represents a naive approach to estimate travel times based on sampled ones. In Figure 4a, dots represent α 's for each sampling run under a given penetration rate, and the dashed line is the average across all 50 runs, both for the linear interpolation approach.

At least for this particular example (Figure 4), if the penetration rate is less than 20%, the linear interpolation approach is superior to the IDE algorithm. However, as the penetration rate increases, the IDE algorithm becomes more effective in estimating delay patterns. If the penetration rate exceeds 40%, this difference is larger than 10%, indicating that the IDE algorithm is significantly better than the linear interpolation approach. Such a trend remains pretty constant as the penetration rate increases further.

The timing of the intersection signal phases was estimated by using the procedure outlined above. This was conducted by using penetration rates ranging from 25% to 100%. Figure 4b depicts the estimated signal phases, with the solid horizontal bars representing the duration of red times. On the top of the figure, the ground-truth signal phases from the simulation are also shown for comparison purposes. The solid vertical lines illustrate the start of red time from the ground-truth signal phases, and the vertical dashed lines indicate the end of red times. At high penetration rates (>60%), the estimated phases are close to the true phases, in terms of both duration of cycles (or red times) and the actual timing. The results, however, deteriorate quickly as the penetration rate becomes smaller.

RESULTS OF FIELD EXPERIMENTS

The PWL intersection delay model and the estimation algorithm were tested by using data from a field experiment. The test site is the intersection of San Pablo Avenue and Solano Avenue in Albany, California (Figure 5a). Data were obtained from two sets of wireless traffic sensors installed upstream and downstream of the sub-

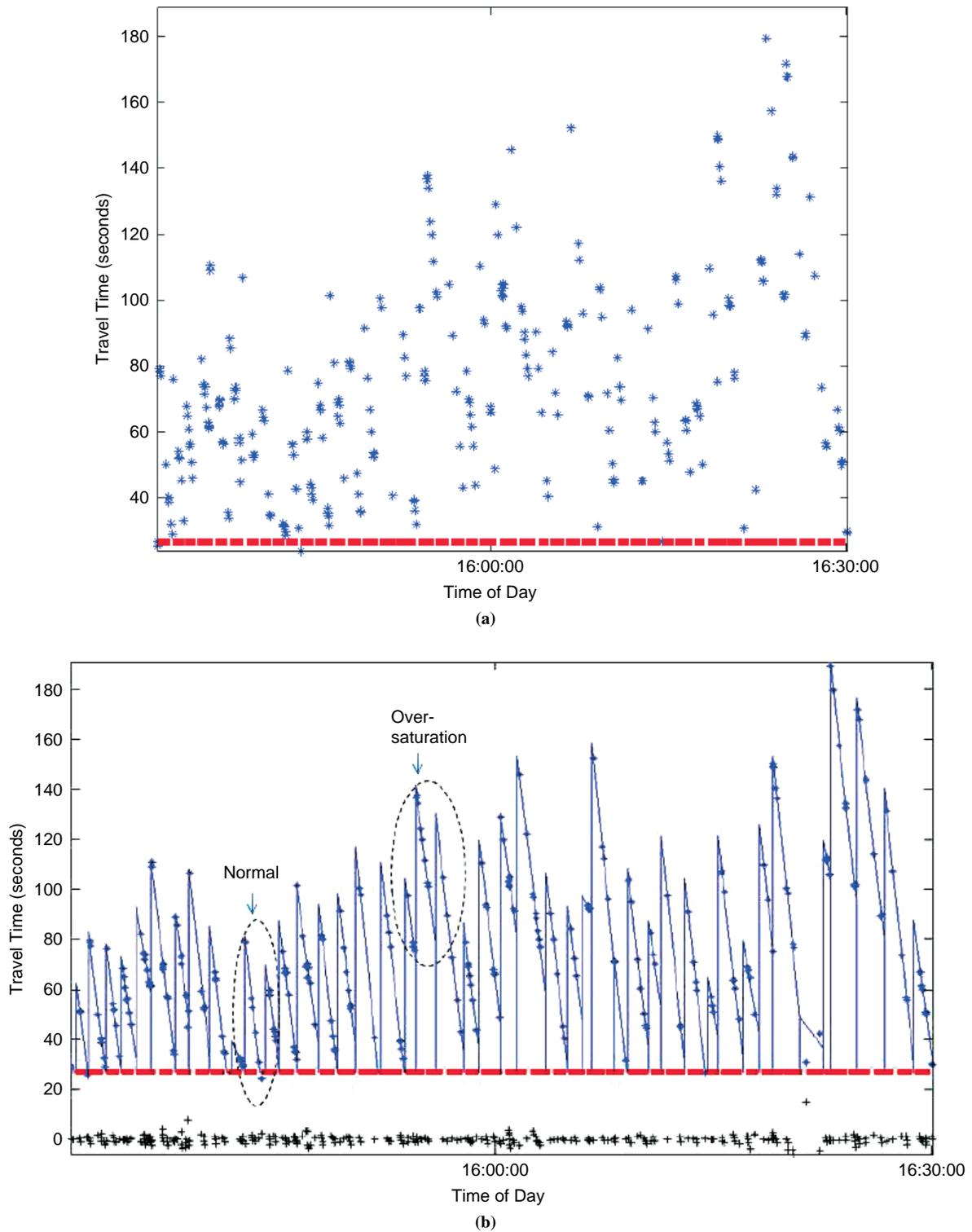
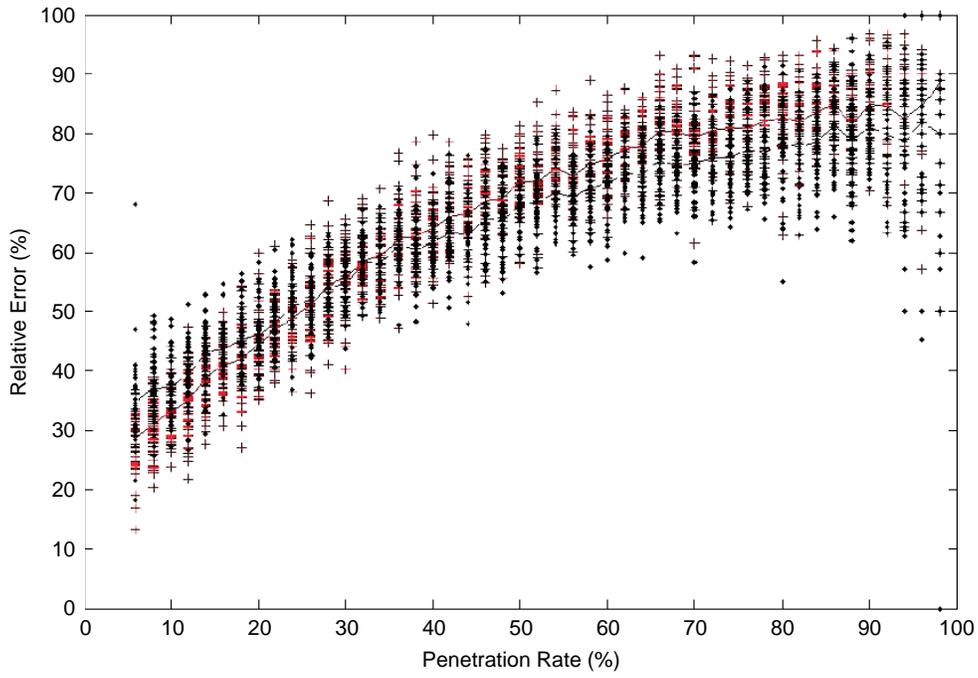


FIGURE 3 Microsimulation results for simulation data for (a) travel times and (b) intersection delay patterns.

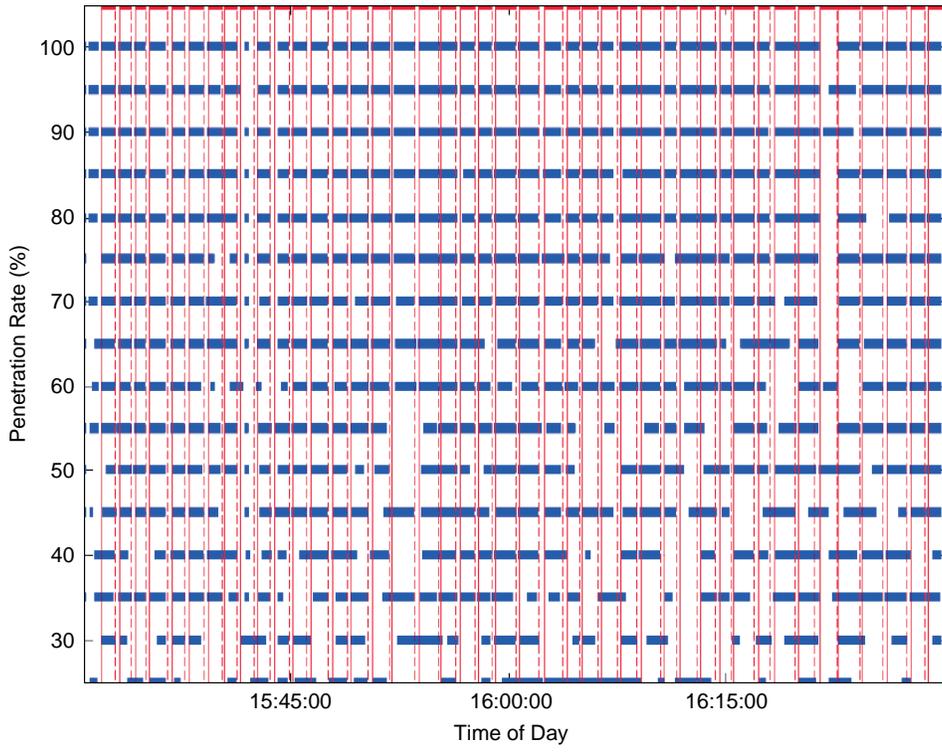
ject intersection. The raw data collected from those sensors contain traffic flow and vehicle signatures. A reidentification algorithm was applied to match vehicles. Travel times between the two sets of detectors were then obtained from the matched vehicles. For detailed descriptions of the test site and the vehicle reidentification algorithm, see Kwong et al. (12). In this article, travel times from

matched vehicles are used directly. In particular, the data contain travel times of 140 vehicles for a 30-min period (1:00 to 1:30 p.m.). Travel time data are shown as asterisks in Figure 5b.

The IDE algorithm described above is applied to the travel times in Figure 5b. The estimated delay pattern curve is shown as thin solid lines. The plus signs represent estimation errors (between

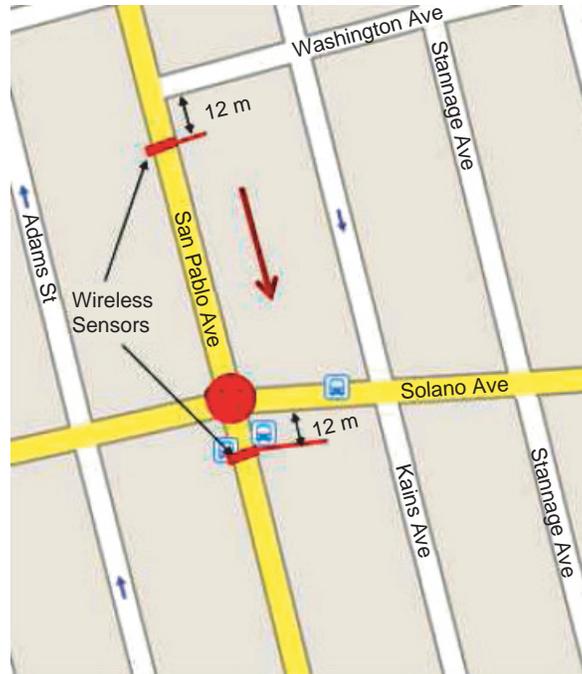


(a)

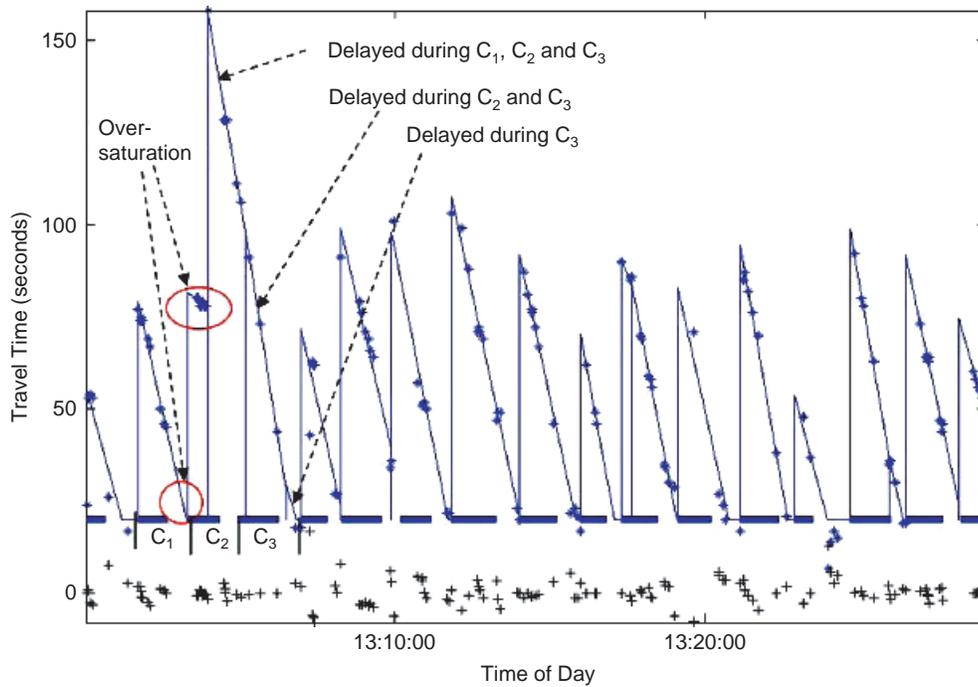


(b)

FIGURE 4 Test results using microsimulation data: (a) impact of penetration rate on delay pattern estimation quality and (b) estimated signal phases.



(a)



(b)

FIGURE 5 Field experiment: (a) test site and (b) estimated delay pattern and signal phases. [Source for (a): maps.google.com.]

asterisks and the delay pattern). For this data set, nearly 88% of vehicles will have an estimation error of less than 15% if the estimated delay pattern is used. This illustrates that the delay pattern is a fairly good estimation to the ground-truth travel times. Three cycles, C_1 , C_2 , and C_3 , during which oversaturation happened are further highlighted in the figure. In fact, oversaturation occurred

during both C_1 and C_2 as marked. During C_3 all queues were cleared up. As a result, the delay pattern for C_3 consists of three line segments as indicated. In particular, the first line segment represents delays caused by all three cycles, the second line segment represents delays caused by C_2 and C_3 , and the third line is for delays caused by C_3 only.

TABLE 1 Estimated Signal Phase Parameters

Index	Cycle Length (s)	Red Time (s)	True Cycle Length (s)	Deviation (%)
1	94	31	108	-13.4
2	98	59	108	-9.4
3	96	61	108	-11.5
4	123	79	108	13.5
5	81	52	108	-25.3
6	114	75	108	5.9
7	97	59	108	-10.2
8	134	88	108	24.0
9	117	71	108	8.4
10	79	50	108	-26.9
11	111	70	108	2.4
12	120	63	108	11.5
13	106	74	108	-1.5
14	107	34	108	-0.9
15	107	79	108	-0.9
16	107	72	108	-0.7

The signal phase estimation procedure described above was also applied to this data set. As this intersection is actuated in coordination with a cycle length of 108 s, Step 4 of the IDE algorithm is applied. The average cycle length generated by the IDE algorithm is 106 s, which is very close to the true cycle length, indicating that at least the cycle-breaking algorithm works fine. The estimated phases (red times) are shown in Figure 5b with thick horizontal bars. Table 1 lists the cycle lengths and durations of red times as well as the deviation (in percent) between the estimated cycle lengths and the true cycle length. Most of the estimated cycle lengths (13 out of 16, or 81%) are within 15% of the true cycle length. The worst case is an underestimate of nearly 27%, while the best case is an underestimate of only 0.7%. This shows that the IDE algorithm proposed in this paper works fairly well for the field experiment data. The estimation algorithms work well mainly because the vehicle reidentification algorithm can match 45% to 65% of total vehicles (12), that is, the penetration rate of the data set is 45% to 65%. As discussed above, such high penetration can generate a reasonable estimation of delay patterns and signal phases.

CONCLUSIONS AND FUTURE PLAN

The authors proposed in this article a two-step algorithm to estimate arterial signalized intersection delay patterns under both normal and oversaturation conditions. First, by investigating the queue-forming and discharging process at signalized intersections, it was shown that intersection delays can be represented as PWL curves. In particular, after the start of the red time, there is always a significant increase in the delay pattern. This unique feature helps to detect the start of a cycle, which in turn makes it possible to break potentially large data samples (i.e., measured travel times) into groups roughly equivalent to signal cycles. A least-squares-based linear fitting algorithm was developed to estimate the delay pattern within a cycle. It was shown that the least-squares method can be converted to solve multiple convex and quadratic programs each with only four variables. Therefore the proposed delay pattern estimation algorithm is polynomial in time and can be implemented in real-time applica-

tions. The model and algorithm were tested by using microscopic traffic simulation data and field experiment data. The results illustrated that the IDE algorithm is promising when the penetration rate is relatively high (e.g., larger than 20%).

The proposed model and algorithm only require sampled travel times obtained between consecutive locations in arterial streets. This is in contrast to most previous intersection delay or travel time models that assume at least signal timing parameters and detector data. As a result, the intersection delay model and algorithm have the potential to be applied in large-scale arterial networks, especially if integrated with the VTL technique designed for GPS-equipped cellular phones. The work presented in this article is only the first step in developing arterial delay models. Some future research directions can be summarized as follows:

1. Only normal and oversaturation conditions were considered in this article. The authors are now working on characterizing delay patterns under other traffic conditions.
2. The least-squares-based IDE algorithm considers only the two most significant features of intersection delays and currently works well for relatively high penetration rates. The algorithm needs to be refined by exploring more characteristics of arterial traffic flow, traffic signal systems, and delay patterns.
3. The model and algorithm were tested by using microsimulation and data from a field experiment. A series of field experiments is currently under way to collect arterial travel times that will be used to test the proposed model. Results will be reported in subsequent articles.

ACKNOWLEDGMENTS

The authors appreciate the discussions with and support obtained from researchers at the Nokia Palo Alto Research Lab, especially Quinn Jacobinson. The authors also appreciate Pravin Varaiya at the University of California, Berkeley, for providing the field experiment travel time data. The first author also thanks Pravin Varaiya and Andy Chow for several insightful discussions on arterial modeling.

REFERENCES

1. Gault, H. E., and I. G. Taylor. *The Use of Output from Vehicle Detectors to Access Delay in Computer-Controlled Area Traffic Control Systems*. Technical report. Research Report No. 31, Transportation Operation Research Group, University of Newcastle upon Tyne, Newcastle upon Tyne, United Kingdom, 1977.
2. Sisiopiku, V. P., and N. M. Rouphail. *Travel Time Estimation from Loop Detector Data for Advanced Traveler Information System Applications*. Technical report. Illinois University Transportation Research Consortium, 1994.
3. Zhang, H. M. Link-Journey-Speed Model for Arterial Traffic. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 1676, TRB, National Research Council, Washington, D.C., 1998, pp. 109–115.
4. Xie, X., R. L. Cheu, and D. H. Lee. Calibration-Free Arterial Link Speed Estimation Model Using Loop Data. *Journal of Transportation Engineering*, Vol. 127, No. 6, 2001, pp. 507–514.
5. Skabardonis, A., and R. Dowling. Improved Speed-Flow Relationships for Planning Applications. In *Transportation Research Record 1572*, TRB, National Research Council, Washington, D.C., 1997, pp. 18–23.
6. Xiong, H., and G. Davis. Field Evaluation of Model-Based Estimation of Arterial Link Travel Times. Presented at 87th Annual Meeting of the Transportation Research Board, Washington, D.C., 2008.
7. Skabardonis, A., and N. Geroliminis. Real-Time Estimation of Travel Times on Signalized Arterials. *16th International Symposium on Trans-*

- portation and Traffic Theory*, University of Maryland, College Park, 2005, pp. 387–406.
8. Liu, H. X., and W. Ma. Virtual Probe Approach for Time-Dependent Arterial Travel Time Estimation. Presented at 87th Annual Meeting of the Transportation Research Board, Washington, D.C., 2008.
 9. Oh, C. *Anonymous Vehicle Tracking for Real-Time Traffic Performance Measures*. PhD thesis. University of California, Irvine, 2003.
 10. Ritchie, S. G., S. Park, S. T. Jeng, and A. Tok. *Anonymous Vehicle Tracking for Real-Time Freeway and Arterial Street Performance Measurement*. Technical report. Research Report, UCB-ITS-PRR-2005-9, California PATH, Berkeley, 2005.
 11. Oh, C., and S. G. Ritchie. Anonymous Vehicle Tracking for Real-Time Traffic Surveillance and Performance on Signalized Arterials. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1826*, Transportation Research Board of the National Academies, Washington, D.C., 2003, pp. 37–44.
 12. Kwong, K., R. Kavalier, R. Rajagopal, and P. Varaiya. Arterial Travel Time Estimation Based on Vehicle Reidentification Using Wireless Sensors. *Transportation Research, Part C*, forthcoming.
 13. S. Amin, et al. Mobile Century—Using GPS Mobile Phones as Traffic Sensors: A Field Experiment. *Proc., 15th World Congress on ITS*, New York, 2008.
 14. Work, D., O. P. Tossavainen, S. Blandin, A. Bayen, T. Iwuchukwu, and K. Tracton. An Ensemble Kalman-Filtering Approach to Highway Traffic Estimation Using GPS-Enabled Mobile Devices. *Proc., 47th IEEE Conference on Decision and Control*, Cancun, Mexico, 2008.
 15. Hoh, B., M. Gruteser, R. Herring, J. Ban, D. Work, J. C. Herrera, and A. Bayen. Virtual Trip Lines for Distributed Privacy-Preserving Traffic Monitoring. *Sixth Annual International Conference on Mobile Systems, Applications and Services (MobiSys 2008)*, Breckenridge, Colo., 2008.
 16. Newell, G. F. *Theory of Highway Traffic Signals*. Institute of Transportation Studies, University of California, Berkeley, 1988.
 17. Roupail, N. M., A. Tarko, and C. Li. Traffic Flow at Signalized Intersections. *Traffic Flow Theory: A State-of-the-Art Report*. Federal Highway Administration, U.S. Department of Transportation, Washington, D.C., 2005. www.tfhrc.gov/its/tft/chap9.pdf.
 18. Chu, L., H. X. Liu, and W. Recker. Using Microscopic Simulation to Evaluate Potential Intelligent Transportation System Strategies Under Nonrecurrent Congestion. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1886*, Transportation Research Board of the National Academies, Washington, D.C., 2004, pp. 76–84.
-
- The Highway Capacity and Quality of Service Committee sponsored publication of this paper.*