Transportation Research Part C 55 (2015) 460–473

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

A real-time adaptive signal control in a connected vehicle environment

Yiheng Feng [*], K. Larry Head, Shayan Khoshmagham, Mehdi Zamanipour

Department of Systems & Industrial Engineering, The University of Arizona, 1127 E. James E. Rogers Way, P.O. Box 210020, Tucson, AZ 85721-0012, United States

ARTICLE INFO

Article history:
Received 15 October 2014
Received in revised form 3 January 2015
Accepted 6 January 2015
Available online 28 January 2015

Keywords:
Adaptive traffic signal control
Connected vehicle
Dynamic programming
Estimating vehicle states
Real-time optimization

ABSTRACT

The state of the practice traffic signal control strategies mainly rely on infrastructure based vehicle detector data as the input for the control logic. The infrastructure based detectors are generally point detectors which cannot directly provide measurement of vehicle location and speed. With the advances in wireless communication technology, vehicles are able to communicate with each other and with the infrastructure in the emerging connected vehicle system. Data collected from connected vehicles provides a much more complete picture of the traffic states near an intersection and can be utilized for signal control. This paper presents a real-time adaptive signal phase allocation algorithm using connected vehicle data. The proposed algorithm optimizes the phase sequence and duration by solving a two-level optimization problem. Two objective functions are considered: minimization of total vehicle delay and minimization of queue length. Due to the low penetration rate of the connected vehicles, an algorithm that estimates the states of unequipped vehicle based on connected vehicle data is developed to construct a complete arrival table for the phase allocation algorithm. A real-world intersection is modeled in VISSIM to validate the algorithms. Results with a variety of connected vehicle market penetration rates and demand levels are compared to well-tuned fully actuated control. In general, the proposed control algorithm outperforms actuated control by reducing total delay by as much as 16.33% in a high penetration rate case and similar delay in a low penetration rate case. Different objective functions result in different behaviors of signal timing. The minimization of total vehicle delay usually generates lower total vehicle delay, while minimization of queue length serves all phases in a more balanced way.

1. Introduction

It has been almost 150 years since the first prototype colored traffic signal light was installed in Westminster, England in 1868 (Webster and Cobbe, 1966). Signal control systems have experienced tremendous development both in hardware and in control strategies. Currently, there are three major traffic control strategies: fixed-time, actuated and adaptive.

Fixed-time control systems utilize historical traffic data to create timing plans for different times of the day (TOD) to address the demand fluctuation. It is assumed that within the entire time period of a particular plan, the traffic demand remains similar. However in reality, the traffic demands may fluctuate quickly. Both actuated and adaptive control strategies

[*] Corresponding author. Tel.: +1 520 248 3988; fax: +1 520 621 6555.
E-mail addresses: yihengfeng@email.arizona.edu (Y. Feng), larry@sie.arizona.edu (K.L. Head), shkhoshmagham@email.arizona.edu (S. Khoshmagham), zamanipour@email.arizona.edu (M. Zamanipour).

http://dx.doi.org/10.1016/j.trc.2015.01.007
0968-090X/© 2015 Elsevier Ltd. All rights reserved.

have been developed to address the problem in real-time. Actuated control collects real-time data from infrastructure-based sensors, e.g. loop-detectors, video detectors, or radar, and applies a simply logic that includes phase calls, green extension, gap out, and max out to make control decisions. Current adaptive control strategies utilize similar real time sensor traffic data to predict near future traffic conditions and seek an optimal timing based on a defined objective function to make control decisions. Several current adaptive signal control systems include ACS-Lite (Luyanda et al., 2003), SCATS (Sims and Dobinson, 1980), SCOOT (Bing and Carter, 1995), OPAC (Gartner, 1983), MOTION (Brilon and Wietholt, 2013), UTOPIA (Mauro and Taranto, 1989) and RHODES (Mirchandani and Head, 2001).

Current adaptive signal control systems rely mostly on data from infrastructure-based sensors, including in-pavement or video based loop detectors. There are two limitations to using the loop based detection. First, loop-detectors are point detectors that only provide instantaneous vehicle location when a vehicle is passing over the detector. There is no direct measurement of vehicle states (location, speed, acceleration) when a vehicle passes the detector. Second, the installation and maintenance cost of the detection system is considered high. If one or more loop detectors are malfunctioning, the performance of the adaptive signal control system can be degraded significantly.

With the advances in wireless communication technology, vehicles are able to communicate with each other (V2V) and with the infrastructure (V2I) through dedicated short range communications (DSRC) and are referred to as connected vehicles. The USDOT suggested that connected vehicle technologies can be applied to areas of safety, mobility and the environment (US DOT Intelligent Transportation Systems Joint Program Office, 2014). Mobility applications utilizing V2I communication enable the intersection to acquire a much more complete picture of the nearby vehicle states. Data from connected vehicles provide real-time vehicle location, speed, acceleration and other vehicle data. Based on this new source of data, traffic controllers should be able to make "smarter" decisions. Collecting connected vehicle data depends on a single infrastructure device (radio) and is significantly less expensive to install and maintain a suite of detectors (e.g. video or loop). If one or more connected vehicles cannot communicate to the intersection due to communication failure, it will only decrease the penetration rate slightly and will have a small impact to the system performance. If the infrastructure based device fails, the intersection returns to the current state of the practice actuated or fixed time control.

The objective of this paper is to present a real-time adaptive traffic control algorithm by utilizing data from connected vehicles. The algorithm is based on improvements of the controlled optimization of phases algorithm (COP) (Sen and Head, 1997) which is used in the RHODES adaptive traffic control system (Mirchandani and Head, 2001). The original COP algorithm was based on a sequence of stages, e.g. A, B, C, D, where a stage could represent phases 1 and 5 (denoted A), or phases 2 and 6 (denoted B), but it did not support flexible, or dual ring, phase sequences. The phase allocation algorithm presented in this paper applies a two-level optimization scheme based on the dual ring controller in which phase sequence and duration are optimized simultaneously.

One of the major problems with the reliance of connected vehicle data for signal control is that the penetration rate of connected vehicles is low, at least for the next few years. Previous research showed that even after federal mandatory installation of DSRC ratio on new light vehicles manufactured in U.S, it may take 25–30 years for connected vehicles to reach a 95% penetration rate (Volpe National Transportation Systems Center, 2008). As a result, in order to provide more accurate arrival data for traffic control, the location and speed of unequipped vehicles needs to be estimated from data from connected vehicles.

The remainder of the paper is organized as follows. Section 2 provides a brief literature review on signal control with connected vehicle data. Section 3 introduces the system framework developed to test signal control applications in a connected vehicles environment in both a real road network and a simulation environment. Sections 4 and 5 describe the phase allocation algorithm and estimating location and speed of unequipped vehicles algorithm, respectively. Section 6 provides test results of the two algorithms based on a VISSIM model. Section 7 closes the paper with conclusions and further research.

2. Literature review

There have been several studies utilizing connected vehicle data for signal control. Different from loop detector data, trajectory data from connected vehicles provide more information of the vehicle states. Utilizing data from connected vehicles, traffic control decisions can be made to be more dynamically responsive to real-time traffic conditions.

Priemer and Friedrich (2009) developed a decentralized adaptive traffic signal control algorithm with V2I communication data. The algorithm was phase based and discretize time into 5 s intervals. The forecast horizon was 20 s. The objective was to minimize total queue length and the problem was solved by dynamic programming and complete enumeration. Goodall et al. (2013) proposed a predictive microscopic simulation algorithm (PMSA) for signal control. The algorithm took data from connected vehicles including position, heading and speed, and then utilized a microscopic simulation model to predict future traffic conditions. A rolling horizon strategy of 15 s was chosen to optimize either delay only or a combination of delay, stops and decelerations. The algorithm considered several market penetration rates and the states of unequipped vehicle were estimated based on equipped vehicle states (Goodall, 2013). However, the algorithm cannot be applied in real-time due to the computational requirements of the parallel simulation to predict the future traffic conditions.

He et al. (2012) proposed a traffic signal control framework for multi modes in a network of traffic signals under V2I environment named PAMSCOD. A headway-based platoon recognition algorithm was developed to identify pseudo-platoons in the network. A mixed-integer linear programming (MILP) problem was solved to find the optimal signal plan based on current traffic conditions, controller status, platoon data and priority requests. Simulation based on a VISSIM model showed that PAMSCOD can consider two modes: transit and passenger cars, and reduce delay for both under-saturated and oversaturated conditions. Result showed that a 40% penetration rate was critical for effectively applying the algorithm. One limitation of PAMSCOD is that the computational requirements increased significantly with the increase in traffic demand since the number of decision variables is proportional to the traffic demand and real-time solution is not currently possible. Under the same V2I framework, He et al. (2014) integrated multi-modal priority control including emergency vehicles, transit buses, commercial trucks and pedestrians, with the consideration of coordination and vehicle actuation. The signal coordination was treated as a virtual priority request in the formulation. However, utilization of traditional vehicle actuation logic within the priority control framework may not be optimal for non-priority vehicles. Lee et al. (2013) presented a cumulative travel-time responsive (CTR) real-time intersection control algorithm with connected vehicle data. The cumulative travel time is defined as the elapsed time from when the vehicle entered the approaching link to the current moment. Travel time included vehicle delay at the intersection. The algorithm applied a Kalman Filter to estimate cumulative travel time under a low market penetration rate. The phasing with highest combined travel time was set to be the next phase. The paper stated that at least 30% market penetration rate is required.

The market penetration rate of connected vehicles is a critical parameter in determining the effectiveness of control algorithms. After the federal mandate for installation of the DSCR device in all new light vehicles manufactured in the US, the connected vehicle penetration rate will steadily increase to a near 100% rate in about 25 years (Goodall, 2013). Some innovative ideas (Fajardo et al., 2011; Lee and Park, 2012) have been proposed based on 100% autonomous vehicle penetration. Under this scenario traffic signals at intersections are removed and the intersection is aware of all vehicle trajectories and given full control of each vehicle. These algorithms were designed to manipulate individual vehicle movements so that vehicles can safely cross the intersection without crashing into each other. Simulation results showed that vehicle delay and travel time can be significantly reduced.

3. System overview

In order to test signal control algorithms under a connected vehicle environment, a testing platform was developed as shown in Fig. 1. This platform supports both real road networks and a simulation environment.

In a real road network, connected vehicles are equipped with on-board equipment (OBE) that is able to broadcast basic safety messages (BSM) through a DSRC radio to the roadside equipment (RSE). The BSM is a message type defined in Society of Automotive Engineers (SAE) J2735 DSRC Message Set Dictionary (Society of Automotive Engineers, 2009) mainly

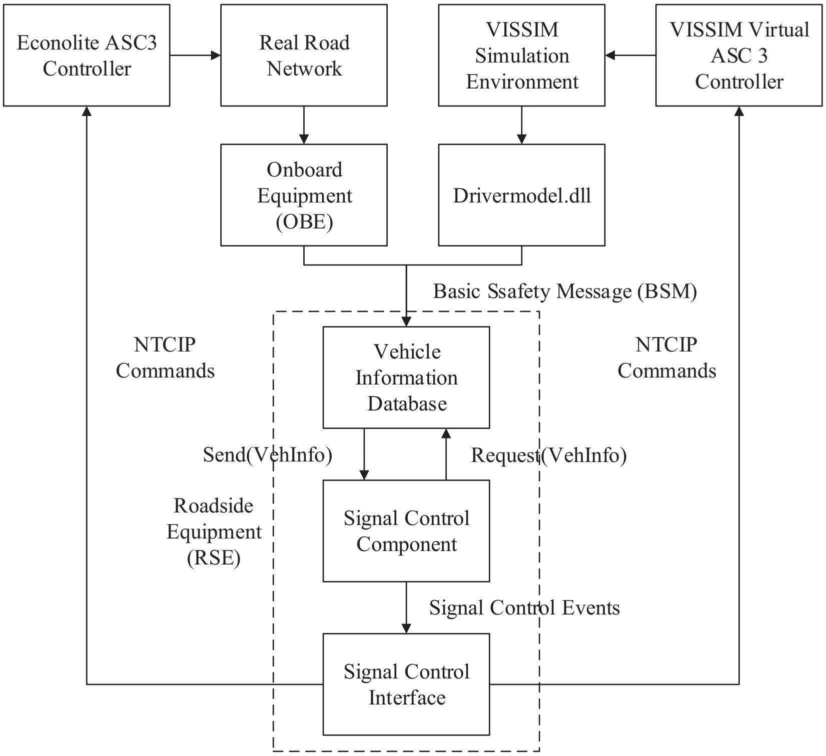


Fig. 1. Signal control under connected vehicle environment system overview.

for safety applications. Each BSM contains vehicle location, speed, heading, vehicle systems information and vehicle size information. Each BSM is broadcast through the DSRC radio at 10 Hz and can be received by the RSE when a vehicle is within radio range of the intersection. In a VISSIM simulation environment, the drivermodel.dll API is used to serve the same BSM broadcast function. There are three components that reside at the RSE: Vehicle Information Database, Signal Control Component, and Signal Control Interface. The Vehicle Information Database receives the BSMs from the OBEs (or VISSIM API) and decodes and saves each vehicle trajectory to the database (temporarily). After the connected vehicle leaves the DSRC range, the vehicle trajectory will be deleted to protect privacy. This component also locates vehicles on the roadway based on a lane-based map and calculates the desired service phase and estimates a time of arrival (ETA) of each connected vehicle.

The Signal Control Component requests vehicle trajectories from the Vehicle Information Database before solving the signal timing optimization problem. The Vehicle Information Database sends the locations and speeds of all the connected vehicles that are within the DSRC range. Upon receiving the data, the Signal Control Component first applies an algorithm to estimate the location and speed of unequipped vehicles. An arrival table is constructed based on the arrival time estimation and serves as the input of the phase allocation algorithm. Detailed descriptions of the two algorithms are provided in Sections 3 and 4, respectively. The output of the phase allocation algorithm is an optimal signal sequence and duration for each phase. The optimal solution is then converted to a list of signal control events and sent to a traffic control interface component. This component controls the signal controller through NTCIP commands (e.g. VEHICLE_CALL, FORCE_OFF, HOLD, and OMIT).

4. Phase allocation algorithm

The phase allocation algorithm assigns signal phase sequences and durations based on predicted vehicle arrivals. The algorithm consists of two levels of optimization. At the upper level, a dynamic program (DP) is applied to each barrier group defined as the collection phases between two barriers of a standard NEMA ring barrier structure as shown in Fig. 2. The formulation can be extended to more general ring barrier controllers, but is limited to the dual ring in this paper. The calculation of the performance function of the upper level is passed to the lower level. The lower level (individual phase) optimization is formulated as a utility minimization problem. The objective can be either minimizing total vehicle delay or queue length based on different operational policies. The sequence of barrier groups is assumed to be fixed, but the order of phases within each ring in each barrier group can vary. Time is discretized to 1 s intervals and the optimization is performed over a predetermined planning horizon, e.g. two cycles.

The following notation is used to define the algorithm:

p: Phase index in each ring and barrier group.
r: Ring index in each barrier group.
j: Index of barrier groups.
T: Total number of discrete time steps in the planning horizon, in seconds.
J: Last stage calculated by the DP before stopping.
x_j: Control variable denoting the length of barrier group j.
s_j: State variable denoting the total number of time steps allocated to barrier group j.
$X_j(s_j)$: Set of feasible control decisions, given barrier group state s_j.
$f_j(s_j,x_j)$: Performance measure at stage j, given barrier group state s_j and control variable x_j.
$v_j(s_j)$: Value function given barrier group state s_j which represents the accumulated performance measure for the current and all previous stages.
$R_{p,r}$: Phase change interval which is the total of the yellow change and red clearance times of phase p in ring r.
$G_{\min}^{p,r}$: Minimum green time of phase p in ring r.
$G_{\max}^{p,r}$: Maximum green time of phase p in ring r.
$X_j^{\min}$: Minimum possible barrier group length of stage j.
$X_j^{\max}$: Maximum possible barrier group length of stage j.
$g_{p,r}$: Green time of phase p in ring r.

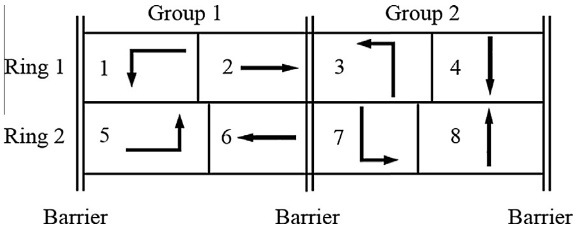


Fig. 2. Ring barrier controller structure with barrier group definition.

n: Planning time index.
$l^{p,r}(n)$: Queue length of phase p at time n in ring r.
$q_a^{p,r}(n)$: Arrival flow rate for phase p at time n in ring r.
$q_d^{p,r}(n)$: Departure flow rate for phase p at time n in ring r.
$d(g_{p,r}, R_{p,r})$: Delay function for phase p in ring r given green time $g_{p,r}$ and phase change $R_{p,r}$.
$A(p, r, n)$: Number of vehicle arrivals for phase p at time n in ring r.

4.1. The upper level

The upper level DP uses a forward and a backward recursion. The forward recursion calculates the performance measure based on the decision and state variables and records the optimal value function for each stage. The backward recursion retrieves the optimal policy starting from the final stage working backwards. The details of the forward and backward recursion are described below.

The forward recursion is based on the allocation of time to each barrier group as stages in the DP. Considering each barrier group as a stage, the algorithm plans as many stages as necessary to get the optimal solution. The ring and phase definition within one barrier group is defined in Fig. 2. The phases in each barrier group are divided into two rings. r represents the ring index and p represents the phase index within the ring. Due to the variability of the traffic demand, the algorithm does not produce a fixed cycle length.

The minimum and maximum allowable barrier group lengths are calculated according to the signal timing parameters as shown in Eqs. (1) and (2). The parameters include minimum green, maximum green, yellow change and red clearance times of each phase. If one phase has no demand, this phase can be skipped which makes $G_{\max}^{1,1}$, $G_{\min}^{p,r}$ and $R_{p,r}$ equal to zero.

$$X_j^{\min} = \max\{G_{\min}^{1,1} + R_{1,1} + G_{\min}^{2,1} + R_{1,2}, \quad G_{\min}^{1,2} + R_{2,1} + G_{\min}^{2,2} + R_{2,2}\} \tag{1}$$

$$X_j^{\max} = \min\{G_{\max}^{1,1} + R_{1,1} + G_{\max}^{2,1} + R_{1,2}, \quad G_{\max}^{1,2} + R_{2,1} + G_{\max}^{2,2} + R_{2,2}\} \tag{2}$$

Given the state variable s_j and the calculated minimum and maximum allowed phase group lengths, the set of feasible control variable can be determined by Eq. (3).

$$X_j(s_j) = \begin{cases} 0 & \text{if } s_j < X_j^{\min} \\ X_j^{\min}, \ldots, X_j^{\max} & \text{if } s_j \geqslant X_j^{\min} \text{ and } T - s_{j-1} - \ldots s_1 > X_j^{\max} \\ X_j^{\min}, \ldots, T - s_{j-1} - \ldots s_1 & \text{if } T - s_{j-1} - \ldots s_1 \leqslant X_j^{\max} \end{cases} \tag{3}$$

The forward recursion is described as follows:

4.1.1. Forward recursion

```
Step 1: Set $j = 1$ and $v_j(0) = 0$;
Step 2: for $s_j = 1, \ldots, T$
          $v_j(s_j) = \text{Min}_{x_j}\{f_j(s_j, x_j) + v_{j-1}(s_{j-1}) | x_j \in X_j(s_j)\}$
Record $x_j^*(s_j)$ as the optimal solution in Step 2.
Step 3: if $(j < 3)$, $j = j + 1$, go to step 2.
          else if $(v_{j-k}(T) = v_j(T))$ for $k = 1$, STOP.
          Else $j = j + 1$, and go to step 2.
```

The forward recursion starts by assigning the first stage to be 1 and the cumulative value function as zero at the beginning of the planning horizon. For each stage, the DP calculates the optimal decision $x_j^*(s_j)$ for each state variable s_j. The objective function $f_j(s_j, x_j)$ in determining the state variable is passed to the lower level optimization with the constraint of control variable x_j. The stopping criteria is achieved if the cumulative value function cannot be improved within two barrier groups (one cycle). The justification of the stopping criterion can be found in (Sen and Head, 1997).

4.1.2. Backward recursion

After all decisions are made in all stages, the optimal decision $x_j^*(s_j)$ of each stage can be retrieved by a backward recursion.

```
Step1: $s_{J-1}^* = T$
Step2: for $\{j = J\text{-}1, J\text{-}2, \ldots, 1\}$
    $s_{j-1}^* = s_j^* - x_j^*(s_j)$
```

The optimal plan is retrieved from stage $J - 1$ since the stopping criteria guarantees the last stage does not improve the solution further.

4.2. The lower level

In Step 2 of the forward recursion, $f_j(s_j, x_j)$, the optimal performance measure at stage j, given barrier group state s_j and control x_j, needs to be calculated. The value of $f_j(s_j, x_j)$ depends on the green duration and phase sequence of the jth barrier group. Different performance measurements can be considered as the objective function. The lower level optimization is formulated in Eqs. (4)–(11).

Given x_j is the length of the barrier group, the lower level problem solves one of the following optimization problems. Objective 4a minimizes the total vehicle delay while objective 4b minimizes the queue length.

$$\min \sum_{r=1}^{2}\sum_{p=1}^{2} d(g_{p,r}, R_{p,r}) \tag{4a}$$

$$\min \sum_{r=1}^{2}\sum_{p=1}^{2} l^{p,r}(s_j + x_j) \tag{4b}$$

s.t.

$$d(g_{p,r}, R_{p,r}) = \sum_{n=s_j+1}^{s_j+x_j} l^{p,r}(n), \quad r = 1,2; \quad p = 1,2 \tag{5}$$

$$l^{p,r}(n) = \max\left[l^{p,r}(n-1) + q_a^{p,r}(n) - q_d^{p,r}(n), 0\right], \quad r = 1,2; \quad p = 1,2 \tag{6}$$

$$q_a^{p,r}(n) = A(p,r,n) \tag{7}$$

$$q_d^1(n) = \begin{cases} \min[q_s, l^{1,r}(n-1) + q_a^1(n)] & \text{if } s_j + 1 \leqslant n \leqslant s_j + g_{1,r} \\ 0 & \text{if } s_j + g_{1,r} < n \leqslant s_j + x_j \end{cases}, r = 1,2 \tag{8}$$

$$q_d^2(n) = \begin{cases} \min[q_s, l^{2,r}(n-1) + q_a^2(n)] & \text{if } s_j + g_{1,r} + R_{1,r} < n \leqslant s_j + g_{1,r} + R_{1,r} + g_{2,r} \\ 0 & \text{if } s_j + g_{1,r} + R_{1,r} + g_{2,r} < n \leqslant s_j + x_j \text{ and } n \leqslant s_j + g_{1,r} + R_{1,r} \end{cases}, r = 1,2 \tag{9}$$

$$\sum_{p=1}^{2}[g_{p,r} + R_{p,r}] = x_j \qquad r = 1,2 \tag{10}$$

$$G_{\min}^{p,r} \leqslant g_{p,r} \leqslant G_{\max}^{p,r} \qquad r = 1,2; \quad p = 1,2 \tag{11}$$

The total vehicle delay (Eq. (5)) is calculated as the summation of queue lengths of all phases in a barrier group during time (x_j). The queue length at time step n (Eq. (6)) depends on the queue length of time step $n - 1$, the arrival flow (Eq. (7)) and departure flow during time step n, which follows the basic flow conservation relationship. The time step is assumed to be 1 s.

Eqs. (8) and (9) consider different phase sequences. Eq. (10) constrains the durations of two phases in each ring should be equal to the current decision variable x_j. In addition, the duration of each phase is bounded by a lower limit and an upper limit which is shown in Eq. (11).

To solve the optimization problem, the combinations of the two phase duration and sequences are enumerated to find the minimum delay combination. This combination is considered as the optimal phase duration and sequence for the given x_j. A rolling-horizon approach is applied where the problem is solved again when one stage (barrier group) is executed in order to include more recent arrival data.

The arrival flow of each phase at each time step ($A(p, r, n)$) comes from a predicted arrival table. The arrival table is a two dimensional matrix with time and phase respectively. The value in each cell is the number of vehicles that will arrive at the stop bar after time step n (ETA) requesting phase p in ring r and is the result of the traffic state estimation algorithm discussed in the following section.

5. Estimating vehicle status of unequipped vehicles

To construct a complete prediction arrival table, the location and speed of each vehicle on the roadway needs to be estimated from the available connected vehicle data. An algorithm called EVLS (Estimation of Location and Speed) serves this function. Fig. 3 describes the general framework of the algorithm.

The road segment of one movement near an intersection is divided into three regions: queuing region, slow-down region, and free-flow region. Black rectangles represent connected vehicles while white rectangles represent unequipped vehicles. In

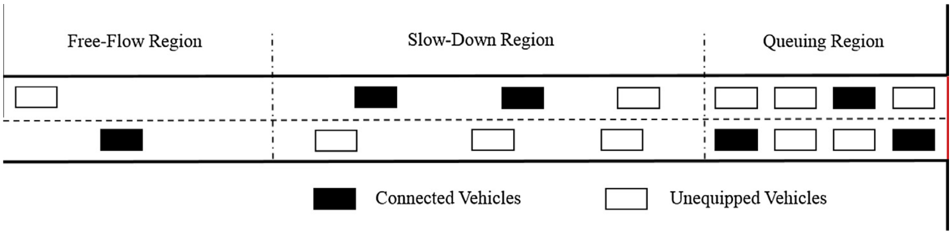


Fig. 3. EVLS algorithm illustration.

practice, the through movement is assumed to have three regions and the left turn movement is assumed to have only two regions because a vehicle can only be identified as turning vehicle after it enters the left-turn bay. Due to the short length of the left-turn bay, it is highly unlikely that the vehicle is still in a free flow state. Different algorithms are applied to each region to estimate the location and speed of each unequipped vehicle.

5.1. Queuing region

If a vehicle is in the queue, the speed is essentially zero leaving the location as the only parameter to estimate. The estimated arrival time of queued vehicles is set to zero in the arrival table, which means queuing vehicles need to be served as soon as possible. Therefore, only the number of queued vehicles is needed to construct the arrival table.

In order to estimate the number of queued vehicles, it is critical to estimate the queue length. Since it is assumed that a queue always begins at the stop bar, the last vehicle in queue needs to be found to determine the queue length.

First, the locations and stopping times of the last stopped connected vehicle and the second to the last stopped connected vehicle in queue are calculated, noted as D_{s1}, T_{s1}, D_{s2} and T_{s2} respectively.

The queue propagation speed v_q between the last two stopped vehicles can be calculated as:

$$v_q = \frac{D_{s1} - D_{s2}}{T_{s1} - T_{s2}} \tag{12}$$

It is possible that some unequipped vehicles may join the queue after the last connected vehicle was observed joining the queue. Because the arrival times of the unequipped vehicles are unknown, the arrival rate (or queue propagation speed) is assumed to remain constant between the arrival of the last connected vehicle and the current time. Assuming the current time is T_c, the estimated queue length l can be estimated by Eq. (13).

$$l = D_{s1} + v_q(T_c - T_{s1}) \tag{13}$$

In some cases, especially under lower penetration rates, there may be only one connected vehicle available within the whole queuing region, the calculation of queue length needs to be adjusted. Assuming the red signal starts at time T_r, and the stopping time of the only connected vehicle is T_{s1}. Then the queue propagation speed can be calculated as:

$$v_q = \frac{D_{s1}}{T_{s1} - T_r} \tag{14}$$

It can be seen that at least one connected vehicle is needed to estimate the queue length. If the average vehicle length is C, the number of vehicles in queue N is then calculated as:

$$N = (\text{INTEGER})l/C \tag{15}$$

Since the actual traffic demand is stochastic and generally nonhomogeneous, it is possible that the last two connected vehicles join the queue in a relative short time but the arrival rate of unconnected vehicles drops significantly afterwards. In this case, if the queue propagation speed is multiplied by the difference between the arrival time of the last connected vehicle and the current time, the queue length can be overestimated. Therefore, Eq. (13) is modified to Eq. (16) to limit the queue propagation time interval.

$$l = D_{s1} + v_q \min\left[(T_c - T_{s1}), \frac{t_a}{Pr}\right] \tag{16}$$

where Pr is the market penetration rate and t_a is a calibrated parameter depending on the flow rate. The queue propagation time interval increases as the penetration rate decreases. The penetration rate of connected vehicles can be computed from the volume (measured using a system detector) and the observed number of connected vehicles over a defined period of time – such as 15 min.

5.2. The slow-down region

The estimation of vehicles in the slow-down region is inspired by (Goodall, 2013). A basic assumption of the estimation algorithm is that the moving vehicles in the slow-down region react with their leading vehicle rationally by some car-following behavior. For simplicity, lane changing behavior is not considered. Wiedemann's car following model is selected to describe the interaction. The Wiedemann car following model classifies the state of a moving vehicle into four stages: free flow, following, closing and emergency (Wiedemann, 1974). The classification is based on the relative position, speed and acceleration of the current vehicle with respect to its leading vehicle. Wiedemann's model contains some calibration parameters that are calibrated based on each of the four stages. The meaning of each parameter is listed in Table 1. Some of the parameters and values are adopted from (Goodall, 2013) while some of them are modified for the estimation algorithm.

The following steps describes the process of the algorithm. The process is repeated for each lane.

Step 1: Identify all connected vehicles in the slow-down region including their location, speed and acceleration.
Step 2: For one pair of adjacent connected vehicles, execute the state determination of the following vehicle as described in the flow chart in Fig. 4(a).
Step 3: Repeat for each pair of connected vehicles in the slow-down region.

The detailed description of step 2 is illustrated below. First, the vehicle state is determined by the classification based on Wiedemann's model as shown in Fig. 4(b). The calculation of the parameters in Fig. 4(b) is based on the parameter definitions in Table 1.

The desired car following acceleration is calculated according to the vehicle state and then is compared to the actual acceleration. Previous research (Goodall, 2013) set the insertion criteria to be based on the actual acceleration being less than the desired acceleration by a predefined threshold, then it is assumed the current vehicle is reacting to a undetected unequipped vehicle. However, vehicles in the slow-down region will not be in the free flow state and the emergency state rarely happens. Following and closing states are more common states for vehicles in the slow-down region as they approach a queue at a traffic signal. Within these two states, the difference between the desired acceleration and actual acceleration seldom reaches the threshold. Therefore, a new criteria for vehicle insertion is proposed for this region: If the vehicle is in the closing state or the following state and the distance between the two connected vehicles is larger than two times the maximum car following distance (*SDX*), it is assumed a vehicle is inserted.

Next, the location, speed and the acceleration of the inserted vehicle needs to be calculated. Suppose the following connected vehicle is the nth vehicle. The inserted vehicle is the $(n-1)$th vehicle. The speed of the $(n-1)$th vehicle depends on the speed and acceleration of the nth vehicle as shown in Eq. (17):

$$v_{n-1} = max\{v_n + \lambda a_n, 0\} \tag{17}$$

Previous research assumed the acceleration rate of the inserted vehicle is always 0 because there is no direct way in Wiedemann's model to calculate the value. This may be fine when the vehicle is in following state (e.g. driving on a free flow section or on a freeway). However, in the slow-down region, the acceleration rate usually is negative indicating the vehicle is seeking to stop. In this paper, if both of the leading $(n-2)$th and following $(n$th) vehicle of the inserted vehicle $(n-1)$th have negative accelerations, then it is assumed that the inserted vehicle will have a negative acceleration as well and the deceleration value is determined by Table 2. The table shows the relationship between the current vehicle speed (v_{n-1}) and the average deceleration rate based on empirical data (Wang et al., 2005). If the criteria is not satisfied, then the acceleration rate of the inserted vehicle is set to be zero.

Table 1
Wiedemann car-following model parameters definition.

Notation	Description	Value used	Unit
τ	Acceleration difference threshold for vehicle insertion	1.96	m/s^2
Δx	Headway	$x_{n-1} - x_n$	m
Δv	Relative velocity	$v_n - v_{n-1}$	m/s
a_{n-1}	Initial acceleration ration for inserted vehicle	Calculated	m/s^2
v	Min speed of vehicle and leader	$\min\{v_n, v_{n-1}\}$	m/s
AX	Minimum headway	6.56	m
BX	Calibration factor	$2.5\sqrt{v}$	–
ABX	Desired minimum headway at low Δv	AX + BX	m
CX	Calibration factor	40	–
EX	Calibration factor	1.5	–
$a_{\max}$	Max acceleration	$3.5–3.5/40v$	m/s^2
$a_{\min}$	Min acceleration	$-20 + 1.5/60v$	m/s^2
SDX	Maximum flowing distance	AX + EX × BX	m
SDV	Decreasing speed difference	$((\Delta x - \text{AX})/\text{CX})^2$	m/s
OPDV	Increasing speed difference	−2.25SDV	m/s
λ	Leading vehicle speed adjustment factor	0.162	–

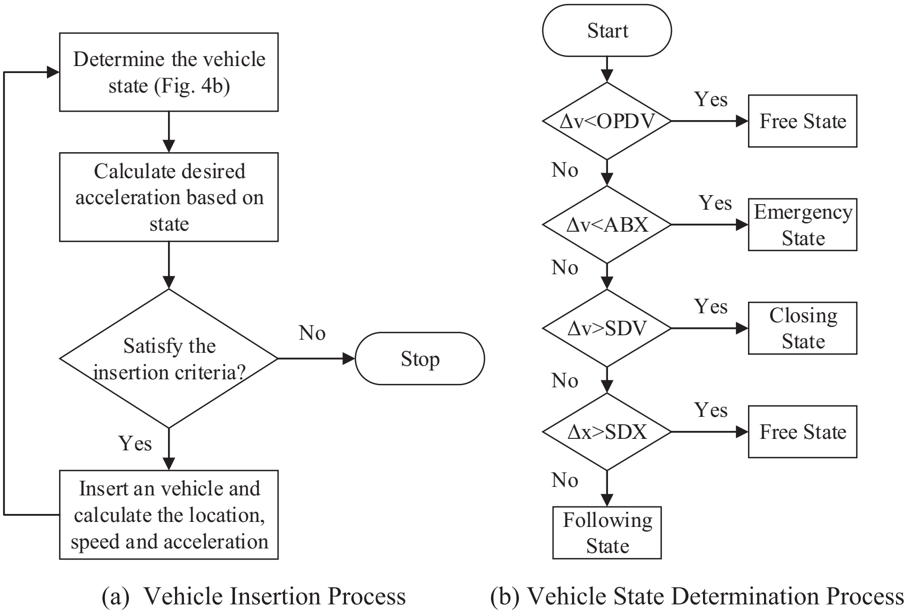

Determine the vehicle state (Fig. 4b)
Calculate desired acceleration based on state
Satisfy the insertion criteria?
No
Stop
Yes
Insert an vehicle and calculate the location, speed and acceleration
(a) Vehicle Insertion Process
Start
Δv<OPDV
Yes
Free State
No
Δv<ABX
Yes
Emergency State
No
Δv>SDV
Yes
Closing State
No
Δx>SDX
Yes
Free State
No
Following State
(b) Vehicle State Determination Process

Fig. 4. Flow chart of the vehicle insertion and state determination process.

By estimating the acceleration of the inserted vehicle, the insertion process can be repeated without further simulation of the vehicle trajectory as in (Goodall, 2013) which cannot be applied in real-time.

If the inserted vehicle is in the closing state, the space headway is estimated by Eq. (18).

$$\Delta x = \mathrm{ABX} - 0.5\frac{(\min\{-\lambda a_n, v_n\})^2}{a_n - a_{n-1}} \tag{18}$$

If the inserted vehicle is in the following state, the position of the vehicle is simply $\Delta x = \mathrm{ABX}$.

The inserted vehicle then becomes a known vehicle and the distance between newly inserted vehicle and the leading connected vehicle is calculated. The process is repeated until no vehicles can be inserted. In the low market penetration rate situation, the distance between two connected vehicles can be very long, therefore the insertion process may be repeated several times.

5.3. The free flow region

The vehicles in the free flow region are assumed to behave independently and not interact with other vehicles. As a result, no car following models is applied to estimate unequipped vehicles in this region. The number of equipped vehicles in all lanes is divided by the penetration rate to get the total number of vehicles The unequipped vehicles can be located uniformly in each lane at random locations on the roadway. The speeds of the unequipped vehicles are assumed to be either the posted speed limit or the observed speed in the field.

Although this estimation is not very accurate, the delay caused by vehicles in the free flow region is small compared to the vehicles in the queuing or slow-down region mainly because these vehicles will not introduce any delay until they arrive at the intersection. On the other hand, vehicles in queuing region introduce delay from the beginning of the planning horizon.

Table 2
Relationship between current speed and deceleration rate.

Deceleration speed interval (km/s)	Deceleration rate (m/s^2)
0–10	0.91
10–20	1.92
20–30	1.82
30–40	1.26
40–50	0.67

6. Results and discussion

To validate the adaptive control algorithm, the intersections of Gavilan Peak and Daisy Mountain in Arizona Connected Vehicle Test Bed (Anthem, AZ) is modeled in VISSIM 6 using the Econolite ASC3 virtual signal controller. The intersection geometry and signal phase diagram are shown in Fig. 5.

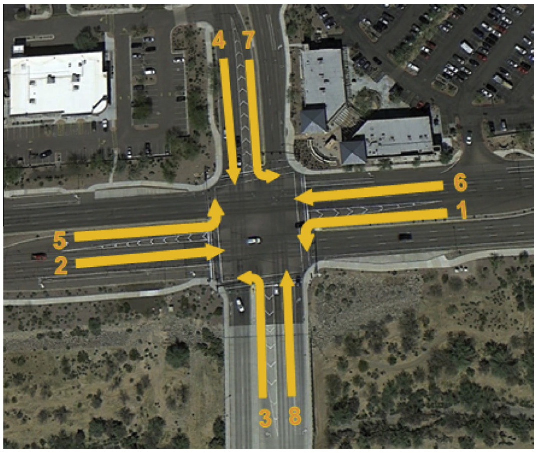
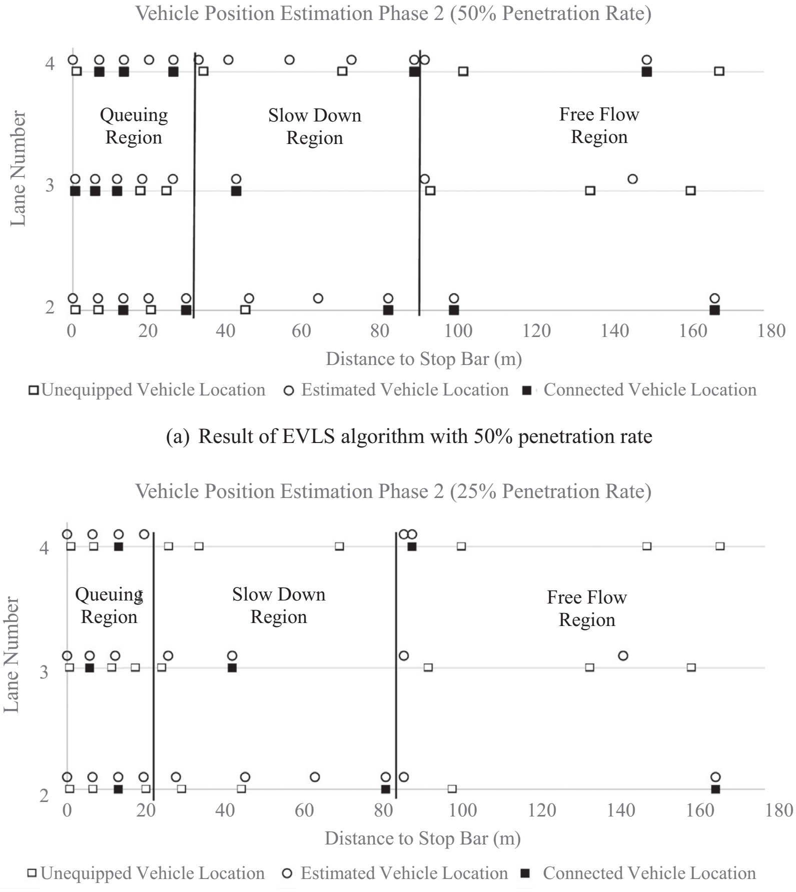


Fig. 5. Geometry and signal phase diagram of Gavilan Peak & Daisy Mountain intersection.



(a) Result of EVLS algorithm with 50% penetration rate

(b) Result of EVLS algorithm with 25% penetration rate

Fig. 6. Estimated locations of EVLS algorithm with different penetration rates.

Table 3
Comparison of average (s/veh) vehicle delay for each phase and total vehicle delay (*s*) under different scenarios (minimization of total vehicle delay).

	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase 6	Phase 7	Phase 8	Total
500/375 demand level									
Actuation	36.22	30.65	34.28	35.39	36.82	27.46	38.19	31.64	39546.26 (0%)
100% PR	43.98	22.85	44.32	26.89	42.12	22.32	37.74	26.90	35576.17 (−10.04%)
75% PR	43.75	23.26	44.36	27.15	36.02	25.50	32.27	34.32	35412.19 (−10.45%)
50% PR	40.22	24.48	40.75	31.25	36.80	23.45	39.20	26.79	35806.88 (−9.46%)
25% PR	40.39	28.37	41.32	30.89	34.95	29.32	42.35	33.62	39874.87 (+0.83%)
667/500 demand level									
Actuation	50.08	38.50	55.96	38.96	57.15	35.13	51.72	37.76	64892.80 (0%)
100% PR	63.22	28.70	64.13	32.74	47.16	27.59	50.20	30.74	55371.99 (−14.67%)
75% PR	61.71	30.89	57.01	34.33	61.01	30.76	43.32	31.96	59004.30 (−9.07%)
50% PR	56.97	31.24	71.61	35.64	45.44	26.80	42.43	33.67	56945.22 (−12.25%)
25% PR	47.98	31.21	55.19	34.47	42.12	32.32	47.85	32.93	57094.73 (−12.02%)

The phase allocation algorithm and EVLS algorithm are executed at the beginning of each barrier group. To ensure a real-time capability, the phase allocation algorithm only plans 2 stages (barrier groups) with an 80s planning horizon. It applies a rolling-horizon approach where the problem is solved again when one stage (barrier group) is executed in order to include more recent arrival data. Before solving the phase allocation algorithm, the EVLS algorithm will be run first to construct a complete arrival table if the market penetration rate is not 100%. The combined processing time of the two algorithms takes about 1 s on an embedded 500 MHz CPU running Linux.

An example of the results of the EVLS algorithm are shown in Fig. 6. This figure shows the estimated vehicle locations versus actual vehicle locations of both connected vehicles and unequipped vehicles of phase 2 (eastbound through movement) with different penetration rates. The vertical line are estimated boundaries of queuing region and slow-down region.

The estimation of vehicles in the queuing region has the best performance which indicates the queue propagation speed and the average vehicle length is estimated accurately. The estimation in the slow-down region tends to insert more vehicles in some cases such as lane 4 with 50% penetration rate because the car-following distance is larger than two times of the maximum following distance, causing the insertion to be triggered. In this case, two more vehicles are inserted based on Wiedemann's car following model. However in reality, different driving behaviors may result in longer car following distances, especially on the approach to a queue at a red traffic signal. On the other hand, in lane four of the 25% penetration rate case, because there is no connected vehicle in the slow-down region, hence no vehicle insertion is triggered. For the free flow region, since the insertion location is randomly selected, the location of the inserted vehicles are not very accurate. However, as discussed before, errors in free flow region have much less impact to the phase allocation algorithm than errors in queuing region. Overall, the performance of the algorithm is satisfying and the estimated vehicle locations show the same pattern as the true vehicle locations.

Based on the results of the EVLS algorithm, the complete arrival table is constructed and used as the input for the phase allocation algorithm. Different scenarios with two different demand levels and four penetration rates are tested. Two demands levels are 500 (667) veh/h/lane eastbound and westbound and 375 (500) veh/h/lane northbound and southbound. Four sets of penetration rates are considered: 100%, 75%, 50%, and 25%. A total of 1125 s are simulated for each scenario which 125 s of warm-up period and 1000 s of data collection time.

Under low penetration rate cases, it is possible that all the vehicles to be served by a certain phase are unequipped vehicles, especially for left turn phases. However, the ELVS algorithm requires at least one connected vehicle in each region. In such situation, the EVLS algorithm will not insert any vehicles and the phase allocation algorithm will skip the phase because of no estimated demand. Those vehicles will wait until one connected vehicle joins the queue. To address the problem, a

Table 4
Comparison of average (s/veh) vehicle delay for each phase and total vehicle delay (*s*) under different scenarios (minimization of queue length).

	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase 6	Phase 7	Phase 8	Total
500/375 demand level									
Actuation	36.22	30.65	34.28	35.39	36.82	27.46	38.19	31.64	39546.26 (0%)
100% PR	33.49	30.37	38.70	29.91	36.47	25.27	31.40	27.63	37027.40 (−6.37%)
75% PR	34.24	28.07	38.89	34.66	33.00	28.97	34.66	30.72	38443.46 (−2.79%)
50% PR	28.22	31.90	33.62	28.01	34.27	25.90	35.82	28.03	36538.61 (−7.61%)
25% PR	35.98	39.87	33.92	33.79	31.50	30.06	33.97	29.77	41675.62 (+5.38%)
667/500 demand level									
Actuation	50.08	38.50	55.96	38.96	57.15	35.13	51.72	37.76	64892.80 (0%)
100% PR	39.44	34.75	41.85	33.51	34.84	28.93	36.26	33.43	54297.80 (−16.33%)
75% PR	54.25	37.51	53.22	35.36	34.87	34.94	43.07	41.21	62140.09 (−4.24%)
50% PR	44.75	34.30	53.36	37.07	37.99	32.95	41.61	34.26	58856.05 (−9.30%)
25% PR	41.60	38.49	52.56	36.97	37.70	42.46	39.47	38.02	64365.98 (−0.81%)

stop-bar loop detector is added to provide information for each left turn phase. The phase allocation algorithm will check the detector to determine if there is no vehicle present and allow the phase to be skipped. If there is no connected vehicle in the lane but the detector places a call to the phase, then the algorithm will not skip the phase and add one vehicle to the queue instead.

The algorithms are run with different objective functions: minimizing total vehicle delay and minimizing queue length. The results of the algorithms are also compared to well-tuned fully actuated control. The unit extension time of the fully actuated control is set to be 1.4 s which is estimated by the recommendation from FHWA (Federal Highway Administration, 2013). Tables 3 and 4 show the performance of the algorithms with different objective functions.

The results show an improvement of the proposed algorithm compared to actuated control when the penetration rate is equal to or greater than 50% in almost all cases. Under the 100% penetration situation, the total delay is decreased by 10.04% and 14.67% under two demand levels when minimizing total vehicle delay, respectively, and 6.37% and 16.33% when minimizing queue length. As the penetration rate decreases, the total delay tends to increase. Under the 500/375 demand level

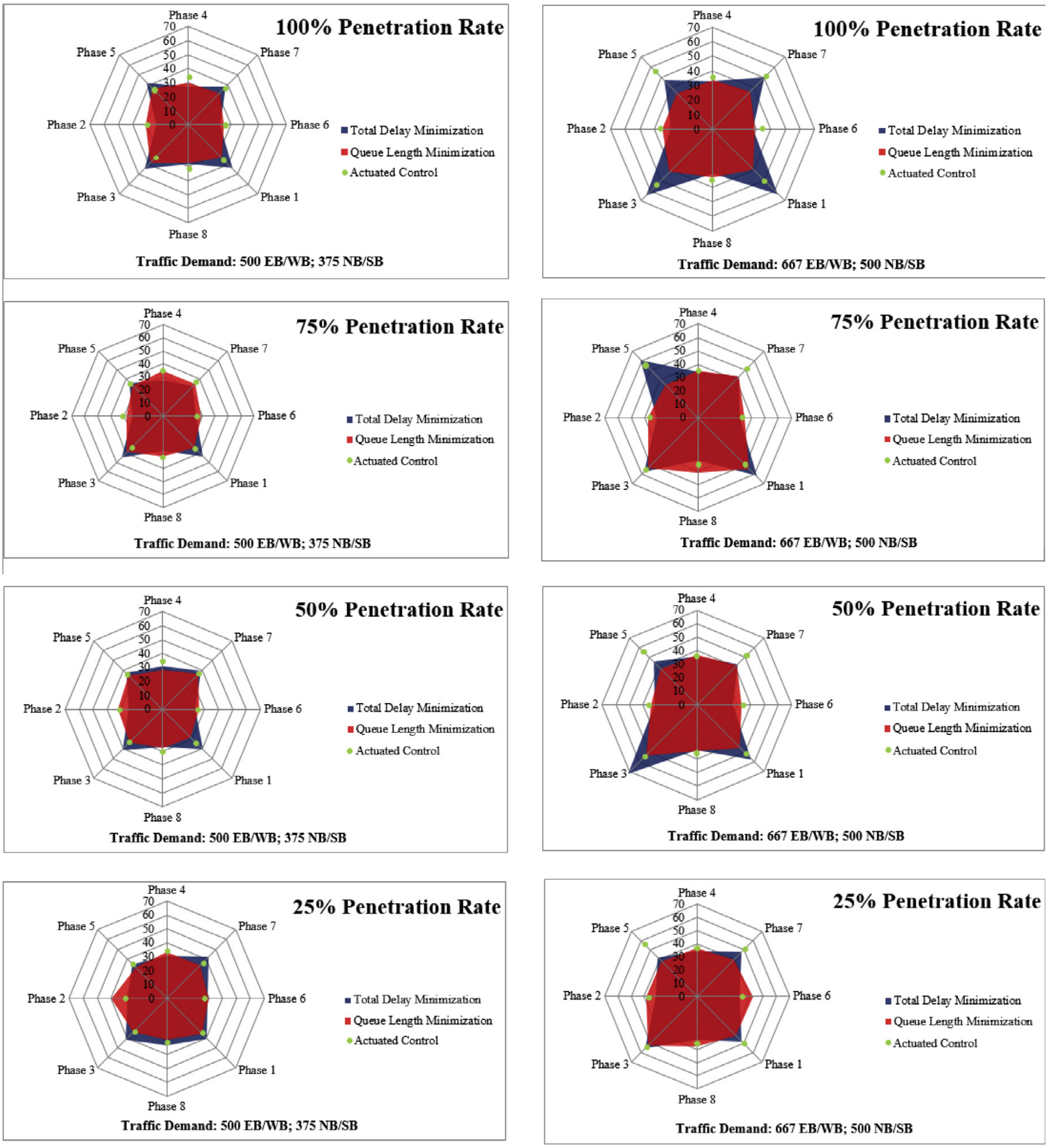


Fig. 7. Comparison of average vehicle delay of each phase under different objective functions.

Table 5
Variance of average delay over all phases.

Penetration rate	100%	75%	50%	25%
500/375 demand level	79.21/17.24	59.23/10.93	46.29/11.56	27.23/9.61
667/600 demand level	202.75/13.59	169.78/55.29	195.24/41.21	71.91/22.59

Note: The values are: Variance of minimizing total vehicle delay/variance of minimizing queue length.

scenario, the performance of the algorithms under the 25% penetration rate assumption have higher total delay than actuated control for both objective functions. However, under the 667/500 demand level, both objective functions outperform actuated control. With the higher demand, the total number of connected vehicle is increased and the estimation errors in the EVLS algorithm are less.

The average vehicle delay of each phase between two objective functions as well as actuated control are compared in a radar diagram (Khoshmagham et al., 2014) as shown in Fig. 7. The radar diagram allows visualization of the performance measures by movement (phase) at the intersection. In a connected vehicle environment, it is possible to directly observe the performance of connected vehicles by movement and by mode.

In general, minimization of total vehicle delay generates lower total vehicle delay compared to minimization of queue length. However, the variance of vehicle delay of each phase is much higher when minimizing total delay as shown in Table 5. The average vehicle delays of left turn phases are much higher than the delays of those through phases. Because the phase allocation algorithm assigns the green duration based on vehicle delay, phases with higher volumes (number of vehicles) receive longer duration and priority which means the phase will be served first within a barrier group. In addition, the algorithm may terminate a left turn phase when the queue has not been fully discharged because the residual queue is not able to introduce enough delay compared to through phases. When the demand is higher, the difference is more significant. On the other hand, when minimizing the queue length, each phase is served more equally, but usually results in higher total vehicle delay. It depends on the policy makers to decide which strategy to implement when operating the signal. For example, if the traffic demand is high, the policy makers can use minimization of total vehicle delay as the objective function to reduce congestion. If the demand is low, minimization of queue length can be used to balance the service of each phase.

7. Conclusions and further work

This paper presents a real-time adaptive phase allocation algorithm for use in a connected vehicle environment. The algorithm utilizes vehicle location and speed data from connected vehicles as the input to optimize phase sequence and duration by solving a two-level optimization problem. Two different objective functions: minimizing total vehicle delay and minimizing queue length are implemented. Since the penetration rate of connected vehicles is not 100%, an algorithm named EVLS was developed to estimate vehicle states of unequipped vehicles base on connected vehicle data.

A real-world intersection was modeled in VISSIM to test the performance and the results were compared to well-tuned fully actuated control. Results showed that the proposed algorithm reduced total delay significantly under high penetration rates and was comparable to actuated control under low penetration rates. Different objective functions result in different behaviors of signal timing. The minimization of total vehicle delay usually generates lower total vehicle delay, while minimization of queue length servers all phases in a more balanced way.

This paper explores how to use connected vehicle data to design an adaptive signal control algorithm. Further research could focus on integrating coordination and priority for special modes, such as transit, trucks, pedestrians, and bicycles into the algorithm. The coordination plan may reduce the flexibility of the phase allocation algorithm because the signals are required to return to coordinated phases at a fixed time point for a certain amount of time. In addition, the current formulation considered the dual-ring barrier structure commonly used in North America. It would be interesting to model other signal control structures, such as stage based controllers, which are widely deployed in European countries.

Acknowledgements

This work has been supported by Arizona Connected Vehicle Initiative, a collaboration between the Maricopa County Department of Transportation SMARTDrive Program, the Arizona Department of Transportation, and the Cooperative Transportation Systems Pooled Fund Multi-Modal Intelligent Traffic Signal Systems (MMITSS) Project.

References

Bing, B., Carter, A., 1995. SCOOT: the world's foremost adaptive traffic control system. In: Traffic Technology International '95'. UK and International Press.

Brilon, W., Wietholt, T., 2013. Experiences with adaptive signal control in Germany. Transport. Res. Rec. J. Transport. Res. Board 2356, 9–16. http://dx.doi.org/10.3141/2356-02.

Fajardo, D., Au, T.-C., Waller, S., Stone, P., Yang, D., 2011. Automated intersection control. Transport. Res. Rec. J. Transport. Res. Board 2259, 223–232. http://dx.doi.org/10.3141/2259-21.

Federal Highway Administration, 2013. Signal Timing on a Shoestring. <http://ops.fhwa.dot.gov/publications/signal_timing/04.htm> (accessed 10.10.14).

Gartner, N., 1983. OPAC: a demand-responsive strategy for traffic signal control. Transport. Res. Rec. 906, 75–81.

Goodall, N.J., 2013. Traffic Signal Control with Connected Vehicles. Ph.D. Dissertation. University of Virginia.

Goodall, N., Smith, B., Park, B., 2013. Traffic signal control with connected vehicles. Transport. Res. Rec. J. Transport. Res. Board 2381, 65–72. http://dx.doi.org/10.3141/2381-08.

He, Q., Head, K.L., Ding, J., 2012. PAMSCOD: platoon-based arterial multi-modal signal control with online data. Transport. Res. Part C Emerg. Technol. 20, 164–184. http://dx.doi.org/10.1016/j.trc.2011.05.007.

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

Khoshmagham, S., Head, L., Saleem, F., 2014. Performance assessment of multi-modal traffic system using micro-simulation methods. In: Presented at the Transportation Research Board 93rd Annual Meeting. Washington, DC.

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

Lee, J., Park, B., Yun, I., 2013. Cumulative travel-time responsive real-time intersection control algorithm in the connected vehicle environment. J. Transport. Eng. 139, 1020–1029. http://dx.doi.org/10.1061/(ASCE)TE.1943-5436.0000587.

Luyanda, F., Gettman, D., Head, L., Shelby, S., Bullock, D., Mirchandani, P., 2003. ACS-lite algorithmic architecture: applying adaptive control system technology to closed-loop traffic signal control systems. Transport. Res. Rec. 1856, 175–184. http://dx.doi.org/10.3141/1856-19.

Mauro, V., Taranto, C.D., 1989. UTOPIA. IFAC Control, Computers, and Communication in Transportation Conference. Paris, France.

Mirchandani, P., Head, L., 2001. A real-time traffic signal control system: architecture, algorithms, and analysis. Transport. Res. Part C Emerg. Technol. 9, 415–432. http://dx.doi.org/10.1016/S0968-090X(00)00047-4.

Priemer, C., Friedrich, B., 2009. A decentralized adaptive traffic signal control using V2I communication data. In: 12th International IEEE Conference on Intelligent Transportation Systems, 2009. ITSC '09. Presented at the 12th International IEEE Conference on Intelligent Transportation Systems, 2009. ITSC '09, pp. 1–6. doi:http://dx.doi.org/10.1109/ITSC.2009.5309870.

Sen, S., Head, K.L., 1997. Controlled optimization of phases at an intersection. Transport. Sci. 31, 5–17.

Sims, A.G., Dobinson, K.W., 1980. The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits. IEEE Trans. Veh. Technol. VT-29, 130–137.

Society of Automotive Engineers, 2009. Dedicated Short Range Communications (DSRC) Message Set Dictionary.

US DOT Intelligent Transportation Systems Joint Program Office, 2014. Connected Vehicle Safety Polot. <http://www.its.dot.gov/safety_pilot/> (accessed 10.10.14).

Volpe National Transportation Systems Center, 2008. Vehicle-infrastructure integration (VII) initiative benefit-cost analysis version 2.3 (draft) (Technical Report). Federal Highway Administration.

Wang, J., Dixon, K., Li, H., Ogle, J., 2005. Normal deceleration behavior of passenger vehicles at stop sign-controlled intersections evaluated with in-vehicle global positioning system data. Transport. Res. Rec. J. Transport. Res. Board 1937, 120–127. http://dx.doi.org/10.3141/1937-17.

Webster, F.V., Cobbe, B.M., 1966. Traffic signals, London Road Research Technical Paper No. 56. H.M.S.O.

Wiedemann, R., 1974. Simulation des strassenverkehrsflusses (Technical Report). University Karlsruhe.