Transportation Research Part C 72 (2016) 109–129

Contents lists available at ScienceDirect

Transportation Research Part C

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

Isolated intersection control for various levels of vehicle technology: Conventional, connected, and automated vehicles

Kaidi Yang [a,*], S. Ilgin Guler [b], Monica Menendez [a]

[a] Traffic Engineering Group, Institute for Transport Planning and Systems, ETH Zurich, Switzerland
[b] Department of Civil and Environmental Engineering, The Pennsylvania State University, University Park, PA 16802, United States

ARTICLE INFO

Article history:
Received 15 February 2016
Received in revised form 7 July 2016
Accepted 16 August 2016
Available online 30 September 2016

Keywords:
Connected vehicles
Automated vehicles
Traffic control
Intersections
Trajectory design
Traffic flow

ABSTRACT

Connected vehicle technology can be beneficial for traffic operations at intersections. The information provided by cars equipped with this technology can be used to design a more efficient signal control strategy. Moreover, it can be possible to control the trajectory of automated vehicles with a centralized controller. This paper builds on a previous signal control algorithm developed for connected vehicles in a simple, single intersection. It improves the previous work by (1) integrating three different stages of technology development; (2) developing a heuristics to switch the signal controls depending on the stage of technology; (3) increasing the computational efficiency with a branch and bound solution method; (4) incorporating trajectory design for automated vehicles; (5) using a Kalman filter to reduce the impact of measurement errors on the final solution. Three categories of vehicles are considered in this paper to represent different stages of this technology: conventional vehicles, connected but non-automated vehicles (connected vehicles), and automated vehicles. The proposed algorithm finds the optimal departure sequence to minimize the total delay based on position information. Within each departure sequence, the algorithm finds the optimal trajectory of automated vehicles that reduces total delay. The optimal departure sequence and trajectories are obtained by a branch and bound method, which shows the potential of generalizing this algorithm to a complex intersection.

Simulations are conducted for different total flows, demand ratios and penetration rates of each technology stage (i.e. proportion of each category of vehicles). This algorithm is compared to an actuated signal control algorithm to evaluate its performance. The simulation results show an evident decrease in the total number of stops and delay when using the connected vehicle algorithm for the tested scenarios with information level of as low as 50%. Robustness of this algorithm to different input parameters and measurement noises are also evaluated. Results show that the algorithm is more sensitive to the arrival pattern in high flow scenarios. Results also show that the algorithm works well with the measurement noises. Finally, the results are used to develop a heuristic to switch between the different control algorithms, according to the total demand and penetration rate of each technology.

* Corresponding author.
E-mail addresses: kaidi.yang@ivt.baug.ethz.ch (K. Yang), iguler@engr.psu.edu (S.I. Guler), monica.menendez@ivt.baug.ethz.ch (M. Menendez).

http://dx.doi.org/10.1016/j.trc.2016.08.009
0968-090X/

1. Introduction

Traffic signals are essential components for urban traffic management. Traffic signals interrupt the progression of vehicles, increasing travel time, gas emissions and fuel consumption (Unal et al., 2003; Coelho et al., 2005; Li et al., 2009). It is estimated that delays at traffic signals contribute to as much as 5–10% of all traffic delay or 295 million vehicle-hours of delay on major roadways in the U.S. (National Transportation Operations Coalition (NTOC), 2012). Moreover, field tests have shown that stop-and-go vehicles cause 14% extra exhaust gas emissions compared to vehicles that drive at a constant speed (Xia et al., 2012).

Traditional signal control strategies use either historical data (fixed-time) or real-time information provided by loop detectors (actuated or adaptive) to determine departure priority. Those devices are usually installed at a fixed location and cannot provide detailed information about the movement of individual vehicles. Therefore, even if those signal control strategies can adapt to the variations in demand, there is still room for improvement.

The recent development in connected vehicle technology (i.e. vehicles that can communicate with each other and infrastructure to provide information on speed, location, etc.) makes it possible to track and control the movement of vehicles and thus has attracted increasing attention in traffic control. With real-time information obtained from connected vehicles, better signal control strategies can be developed. Moreover, wireless communication systems and automated driving can help advise drivers or control vehicles, providing a more flexible design of signal control strategies.

A recent research (Guler et al., 2014) proposed a signal control algorithm for an isolated intersection. It used information provided by different penetration rates of connected vehicles present in a traffic stream, and evaluated the benefits of this technology. This paper further exploits the value of connected vehicle technology by extending the above research. In this work, additionally, a certain percentage of connected vehicles are assumed to be automated. This allows the central controller to optimally design the trajectories of these automated vehicles to further improve the operations. Compared to Guler et al. (2014), the contributions of this paper are threefold. First, this work proposes a scheme to show how the algorithm should evolve with different development stages of connected vehicle technology, considering three different types of vehicles. Second, it enables bidirectional vehicle-to-infrastructure communication and successfully integrates trajectory design for automated vehicles into the signal control scheme. Third, it further improves the performance indices by reducing both the delay and the number of stops.

This paper is organized as follows. Section 2 presents a short review on signal control strategies utilizing connected vehicle technology. Section 3 introduces the algorithm developed. Section 4 evaluates the performance of this algorithm by comparing it to an actuated signal control algorithm, and analyzes its robustness. Section 5 proposes a demand responsive control strategy based on the application bounds of the new algorithm. Section 6 concludes the paper.

2. Literature review

This section presents a short literature review on signal control strategies based on connected vehicle technology. The interested readers can refer to Florin and Olariu (2015) for a comprehensive survey of traffic signal optimization methods using connected vehicles from a technology perspective, and L. Li et al. (2014) which, from the control side, summarized the general traffic control strategies and highlighted the transition from feedback control to feed-forward control thanks to the connected vehicle technology.

The existing research on intersection control using connected vehicles can be classified into two categories. In the first category, it is assumed that vehicles report position and speed information to a central controller. The central controller then optimizes the intersection control based on such information. Some studies focus on the optimization of the signal phases (Priemer and Friedrich, 2009; Hu et al., 2015), whereas other studies provide priority to individual cars to optimize departure sequences (Wu et al., 2007; Pandit et al., 2013). Most of the early works assume all or a majority of the vehicles are connected. Only a few recent works relaxed this assumption by taking into consideration incomplete information. The arrival information of unequipped vehicles is estimated using either traffic models (He et al., 2012, 2014; Guler et al., 2014; Feng et al., 2015), statistical methods (Lee et al., 2013), or simulations (Goodall et al., 2013). It is shown that the algorithms in the aforementioned works perform well with lower penetration rates. However, the benefit of this technology is not fully exploited in this category, as these works assume only uni-directional communications.

The second category takes advantage of automated driving and integrates trajectory design into the signal control scheme. Vehicle trajectories can be designed to minimize evacuation time (Li and Wang, 2006), or to provide cooperative control (Lee and Park, 2012). A reservation based algorithm that reserves space at the intersection for each car in advance was proposed for automated vehicles only (Dresner and Stone, 2004) and for connected vehicles with human drivers mixed with automated vehicles (Dresner and Stone, 2006). Au and Stone (2010) presented a trajectory planning algorithm for automated vehicles to reduce the number of stops. Z. Li et al. (2014) presented an on-line algorithm to optimize vehicle trajectory and traffic signal simultaneously for an intersection of two one-way streets. A rolling horizon scheme is adopted to identify each control stage. In each stage, trajectory is designed and signal timing plans are enumerated to minimize total delay. Kamal et al. (2015) used model predictive control to coordinate automated vehicles at an unsignalized intersection to maximize the intersection capacity and avoid collision. One limitation in the previous works in this category is that they assume all vehicles to be automated or connected.

This paper tries to fill the gap. First, this paper allows the central controller to design the trajectory of automated vehicles to improve the overall operations. Second, this paper considers incomplete information. Finally, a demand responsive scheme is proposed to accommodate all stages of technology development.

3. Description of algorithm

In this paper, a signal control algorithm is developed using the information obtained from connected and automated vehicles at an isolated intersection with two one-way traffic streams. This simple scenario is chosen to gain insights on the benefits of connected vehicle technology for the operation of the intersection. For the reader's convenience, a list of the most important variables used in the algorithm is presented in Table 1.

Three categories of vehicles are considered: (1) conventional vehicles, (2) automated vehicles (also known as autonomous vehicles in some literature), and (3) connected vehicles. Conventional vehicles are defined as the vehicles that cannot communicate to the central controller by any means. Automated vehicles are the vehicles without a driver, whose trajectories can be controlled by the central controllers. Although all automated vehicles provide information (i.e., are connected), herein we call connected vehicles only those vehicles that send information, but whose trajectory cannot be controlled/modified.[1]

It is assumed that both connected and automated vehicles are equipped with vehicle-to-infrastructure (V2I) communication systems. V2I communication systems have been widely employed in traffic signal control strategies (He et al., 2014; Z. Li et al., 2014; Hu et al., 2015). There are several platforms that enable V2I communications, such as Dedicated Short Range Communications (DSRC), 3G/4G, and Bluetooth. Compared to the other platforms, DSRC has several advantages (Guo and Balon, 2006; Andrews and Cops, 2009): (1) it is more robust to radio interference and extreme weather conditions; (2) it works with high vehicle speeds; (3) it has small latency (0.002 s); (4) it helps to protect privacy; and (5) the transmission rage of DSRC is 100–1000 m. Therefore, the algorithm in this paper is developed based on the DSRC communication technology.

Additionally, it is assumed that the trajectory (vehicle motion trajectory, e.g. speed, acceleration rate, etc.) of automated vehicles can be controlled using V2I communications and automated driving systems. This gives the possibility to design the trajectory for automated vehicles to improve the traffic operations at the intersection.

The proportion of each type of vehicle is summarized with two penetration rates. The percentage of connected vehicles among all vehicles is defined as the information level, i.e.,

$$\text{information level} = \frac{\text{number of automated vehicles} + \text{number of connected vehicles}}{\text{number of all vehicles}} \tag{1}$$

The proportion of automated vehicles among all vehicles that send information (connected and automated vehicles) is defined as the automated level, i.e.,

$$\text{automated level} = \frac{\text{number of automated vehicles}}{\text{number of automated vehicles} + \text{number of connected vehicles}} \tag{2}$$

Both the information level and the automated level range from 0 to 1. Different combinations of these two penetration rates represent different implementation stages of connected vehicle technology.

The inputs to the algorithm include the car set N that consists of all connected and automated cars in the zone of interest, plus stopped conventional cars ahead of some stopped connected or automated cars, as well as the arrival sequence and the observed or estimated trajectory of each car. The output of the algorithm is the optimal departure sequence for all cars in car set N and the optimized trajectory for each automated car in the zone of interest. The zone of interest is defined as the range within which connected and automated vehicles can send and receive information in relation to the intersection. The length of the zone of interest partly determines the amount of information received from the connected and automated vehicles. It is constrained by two factors: the city block size (traffic side) and the DSRC transmission range (technology side). It would be unrealistic to have a zone of interest longer than a city block, as this would require a priori information on route choice. It is also required for the zone of interest to be shorter than the DSRC transmission range due to physical constraints. However, if the zone of interest is selected to be too short, there will not be enough information resulting in myopic signal control. Therefore, the minimum of the city block size and the DSRC transmission range is used in this paper.

The flowchart of the algorithm is shown in Fig. 1. The goal of this algorithm is to determine the optimal departure sequence of all vehicles and the optimized trajectories for automated vehicles in an on-line manner (i.e. in real time). Each decision step (i.e. the time between every update of the optimal departure sequence and trajectories of automated vehicles) is triggered by one of three events: (1) a connected or automated car enters the zone of interest; (2) a connected or automated car comes to a stop; (3) a connected or automated vehicle leaves the zone of interest. All of these events require the inputs to be updated. The first event adds new connected or automated cars to the set N, the second event unveils stopped conventional cars that were not detected before, and the third removes a vehicle from the car set N.

[1] In cases where connected vehicles also receive instructions, if the drivers follow them, they are deemed as automated vehicles; if the drivers do not change their behaviors accordingly, they fall into the connected vehicle category. For simplicity, the case where the drivers react to the instructions but do not follow them exactly is not considered in this paper.

Table 1
List of most important variables.

N	Current car set at each decision time, cars indexed by c
I	Arrival sequence, cars indexed by i
J	Departure sequence, cars indexed by j
M	Approach, indexed by m
u_f	Free flow speed
k_{jam}	Jam density
w	Backward wave speed
S_m	Saturation flow rate for approach m
l	Length of the intersection
a	Acceleration rate
d	Length of zone of interest
$u_{\min}$	Lower bound for speed in trajectory design
V_c	Virtual departure time of car c at the downstream end of the intersection
$D_{c,J}$	Departure time of car c for departure sequence J at the downstream end of the intersection
TD_J	Total delay for departure sequence J
$P_{c,J}$	Delay penalty for car c for departure sequence J, this represents the time it takes for car c to cross the intersection
$O_{c,J}$	Position of car c within the platoon using departure sequence J
$u_{c,J}^{init}$	Initial speed of car c when entering the intersection
$u_{c,J}^{opt}$	Optimal speed in trajectory design
$u_{c,J}^{des}$	Design speed in trajectory design

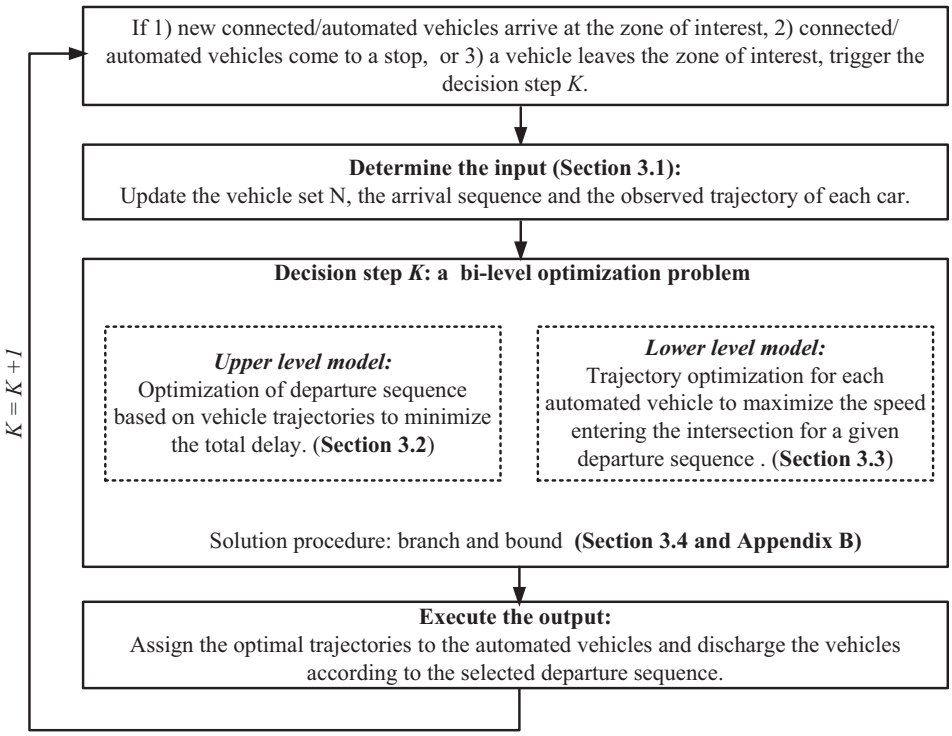


Fig. 1. Flowchart of algorithm.

In each decision step, a bi-level optimization model is proposed to integrate the optimization of departure sequence and the trajectory design for automated vehicles based on the arrival information. The upper level model is adapted from Guler et al. (2014). In the upper level model, the departure sequence is optimized to minimize the total delay based on the revealed vehicle trajectories. In the lower level model, the trajectory of each automated vehicle is optimized for a given departure sequence to maximize the speed entering the intersection. The two levels interact with each other. The lower level model takes the departure sequence and returns vehicle trajectories to the upper level model. In this paper, the bi-level model is solved by a branch and bound algorithm.

In the final step, vehicles are discharged according to the calculated departure sequence. Automated vehicles will be given designed trajectories. It is assumed that the signal turns green for an approach right before the first car in that platoon enters the intersection.

The rest of this section presents the algorithm within each decision step. Section 3.1 presents how the model inputs are determined. Section 3.2 presents the upper level model, i.e. optimization of departure sequence. Section 3.3 presents the lower level model, i.e. trajectory design for automated vehicles. Section 3.4 proposes a fast branch and bound algorithm to solve the bi-level model.

3.1. Model inputs

When the decision step is triggered, the inputs to the algorithm, namely the vehicle set, arrival sequence and the trajectory of each car need to be updated. The method to update automated and connected vehicles differs from that for conventional vehicles.

The arrival sequence and trajectories of connected and automated vehicles are obtained directly from the V2I communication systems. It is assumed that connected vehicles and automated vehicles report their location and speed (i.e. trajectory) at any given time after they enter the zone of interest. With such information and assuming free flow speed u_f, the virtual departure time V_c (i.e. the time at which a vehicle would arrive to the downstream end of the intersection if there were no queueing) of an automated or connected vehicle c can be obtained and updated once it enters the zone of interest.

The arrival information (arrival sequence and virtual departure time) of conventional vehicles (i.e. vehicles that do not provide any type of information) can be estimated using the location information of connected and automated vehicles. Once a connected or automated vehicle comes to a stop, the number of conventional vehicles stopping in front of it (i.e. the arrival sequence of conventional vehicles) can be estimated by kinematic wave theory (given previous signal settings and assuming a fundamental diagram). The detailed procedure for estimating the number of conventional vehicles can be found in Appendix A. The virtual departure times for these conventional vehicles are estimated using a linear interpolation.

3.2. Upper level model: optimization of departure sequence

Denote M as the set of approaches. Denote I as the arrival sequence, i.e. the sequence in which vehicles on both approaches in car set N arrive at the zone of interest. For automated and connected vehicles, the arrival sequence and approach are detected from the car-to-infrastructure system once they enter the zone of interest. The arrival sequence and approach for conventional vehicles are estimated using the information provided by connected and automated vehicles.

Denote J as the departure sequence, i.e. the sequence in which vehicles on both approaches in car set N depart from the intersection. Note that departure sequence J is one permutation of the arrival sequence I that satisfies the first-in-first-out principle for each individual approach. To minimize total delay, the optimal departure sequence is found, i.e.,

$$\min_J TD_J \tag{3}$$

For each departure sequence J, the total delay TD_J is calculated as

$$TD_J = \sum_{c \in N} (D_{cJ} - V_c) \tag{4}$$

where for departure sequence J, D_{cJ} represents the predicted departure time of vehicle c from the downstream end of the intersection. Recall that V_c is the virtual departure time from the downstream line of the intersection for car c. Hence, the difference $D_{cJ} - V_c$ is the delay for car c.

Assume each vehicle is represented by a triple $c = (i, j, m)$ where $i \in I$ is the position of car c in the arrival sequence, $j \in J$ is the position of car c in the departure sequence, and $m \in M$ is the approach. For a given departure sequence J, and the corresponding optimal trajectories obtained in Section 3.3, the predicted departure time for car $c = \{i, j, m\}$, D_{cJ}, is calculated as the maximum of the virtual departure time and the next possible departure time, i.e.[2]

$$D_{cJ} = \max\left\{V_c;\ D_{c'J} + \frac{1}{S_m} + P_{cJ}\right\}, \quad c = \{i, j, m\} \text{ and } c' \in \{I, j-1, M\}, \quad \forall c \in N \tag{5}$$

where S_m is the saturation flow from approach m, c' is the previous vehicle in this departure sequence to vehicle c. The delay penalty P_{cJ} represents the time it takes for each car to cross the intersection, and is given in Eq. (6) derived with the basic kinematic law.

$$P_{cJ} = \max\left\{\frac{l}{u_f};\ \frac{-u_{cJ}^{init} + \sqrt{\left(u_{cJ}^{init}\right)^2 + 2al}}{a}\right\} \tag{6}$$

[2] For the very first vehicle entering the system when the system is empty, we expect that it drives at the free flow speed, thus we have $D_{cJ} = V_c$. For the first vehicle in the departure sequence J which is not the first vehicle in the system, Eq. (5) still holds, where c' is the previous vehicle that departs before vehicle c (although c' is not in the departure sequence J but in the departure sequence of an earlier time step).

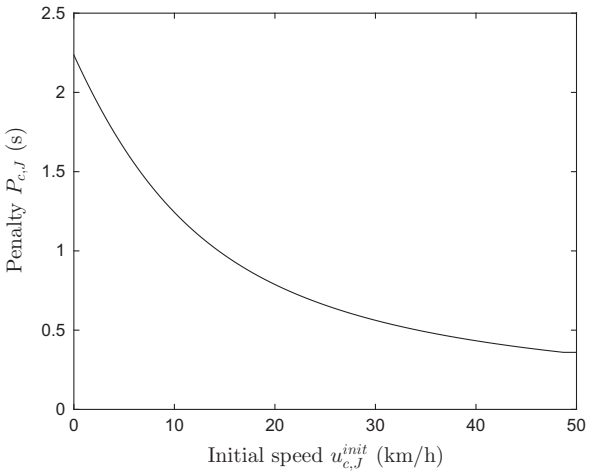


Fig. 2. Illustration of penalty, where $u_f = 50$ km/h, $l = 5$ m, $a = 2$ m/s^2.

where l is the length of the intersection; a is the acceleration rate, and $u_{c,J}^{init}$ is the initial speed of a car c when entering the intersection. The initial speed $u_{c,J}^{init}$ is the result of the trajectory optimization for automated vehicles (more details are provided in Section 3.3). The second term of the RHS in Eq. (6) represents the acceleration time of car c if the initial speed is smaller than the free flow speed u_f. Here, the case where the vehicle reaches the free flow speed in the middle of the intersection is not considered. This might cause a small and systematic error which should not influence the results in any significant manner.

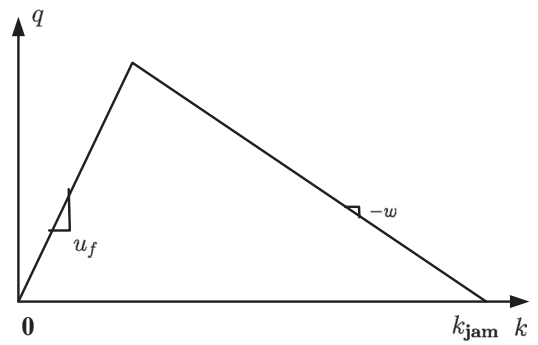
As is shown in Fig. 2, the penalty decreases as the initial speed increases. Notice that as more cars discharge from a platoon, the initial speed increases since cars have space to accelerate before consecutively entering the intersection. Therefore, the penalty favors platooning instead of alternating departures between approaches. By not requiring cars to stop, the initial speed can also be increased. Hence, allowing cars to discharge in platoons is favorable for decreasing the penalty.

3.3. Lower level model: trajectory design for automated vehicles

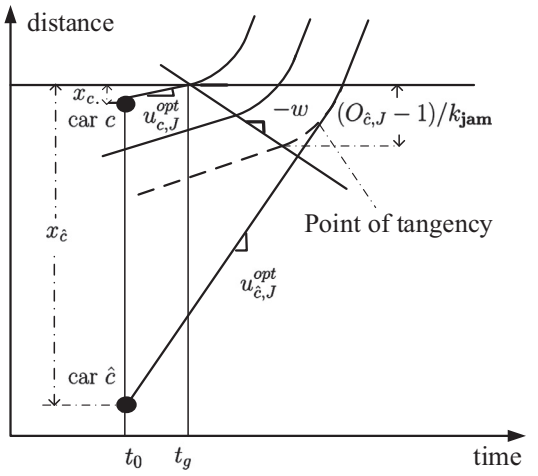
For a given departure sequence J, the trajectory of each vehicle can be sequentially determined according to the departure sequence. The trajectory is designed for each individual automated vehicle based on the real or estimated traffic information (departure time, speed, etc.) of the cars in front of it. The objective of the trajectory design is to let vehicles pass the intersection at a specific time with the maximum possible speed and, if possible, without stopping.

In order to calculate the optimal trajectories for automated vehicles, the "future" trajectories of connected and conventional vehicles should be predicted using Newell's car following model. One input to the Newell's car following model is the departure sequence, which is represented as the signal timing. The other parameters of the Newell's car following model are determined using kinematic wave theory. For Newells car following model the distance travelled in a time step is calculated as the minimum of how far a vehicle can travel at free-flow speed and what would be permissible due to downstream congestion. Under free-flow conditions, the trajectory of the car no longer depends on the car in front of it. In the case with downstream congestion, the spacing between each pair of cars is the inverse of the density. For more information on Newell's car following model, see Newell (2002).

For ease of presentation, the acceleration and deceleration processes are simplified without changing the overall control. To this end, the acceleration and deceleration processes are divided into two types. The first type occurs because the signal changes from red to green. This type of maneuver includes the vehicles that accelerate from the stopped state and cross the intersection (see Fig. 3b for an example). This type involves only acceleration maneuvers and usually occurs close to the intersection. The second type of maneuver consists of all other possibilities, i.e. either decelerating because of red signal or a queue downstream, or informed by the controller to accelerate or decelerate. This type involves both acceleration and deceleration maneuvers and occurs near the back of queue, which can be near the intersection or not. In other words, the first type consists of the accelerating vehicles because the signal switches to green and the second type includes all the other cases. The first type of acceleration process is taken into consideration to properly evaluate the benefits of platooning (see Eq. (6)). For simplicity, constant acceleration with acceleration rate a is assumed. For the second type, instantaneous acceleration and deceleration are assumed to simplify the discussion (i.e. $a = \infty$). Notice that this assumption of instantaneous acceleration and deceleration can be made without loss of generality. As long as the time to enter the intersection is fixed, the assumption of instantaneous acceleration and deceleration will only slightly influence the initial speed $u_{c,J}^{init}$ at



(a) Triangular fundamental diagram



(b) Time space diagram with sample trajectories

Fig. 3. Illustration of the speed calculation. Car c represents the first car in the platoon; car $\hat{c}$ represents the car that follows other cars. The dashed trajectory represents the closest possible trajectory of car $\hat{c}$ to the last vehicle in the platoon (according to Newell's car following model).

which the vehicle enters the intersection, and thus change the total calculated delay by Eqs. (5) and (6). However, as is shown in Fig. 2, the marginal change in delay penalty decreases as $u_{c,J}^{init}$ increases. Hence, as long as $u_{c,J}^{init}$ is sufficiently large (i.e. over 25 km/h), the total calculated delay across all cars will only be slightly different than in reality. Therefore, it is reasonable to assume instantaneous acceleration and deceleration before a car joins the dispatching platoon or comes to a stop.

By the assumption above, the design of the full trajectory of an automated vehicle c can be simplified into deciding on a particular design speed $u_{c,J}^{des}$ at the time the algorithm performs trajectory design. This is a quasi-optimal solution, which simplifies the discussion with satisfactory approximation. The initial speed entering the intersection $u_{c,J}^{init}$ can be calculated from $u_{c,J}^{des}$.

Intuitively, it is always possible to let a car pass the intersection without stopping, as long as the car drives at a sufficiently low speed. However, it does not make sense if the car crawls to the intersection. Therefore, a lower bound $u_{\min}$ is defined to constrain the design speed, i.e.,

$$u_{\min} \leqslant u_{c,J}^{des} \leqslant u_f \tag{7}$$

where the second inequality means that the design speed cannot exceed the free flow speed u_f.

Given the trajectory of the previous cars, an automated car can be assigned to either wait until the next signal cycle, or follow the previous car. This is equivalent to considering two cases based on whether an automated car is the first car in its platoon or not (this is determined from the departure sequence), i.e. $O_{c,J} = 1$ or $O_{c,J} > 1$ where $O_{c,J}$ represents car c's position in the platoon for a given departure sequence J.

As is shown in Fig. 3b, car c represents the case with $O_{c,J} = 1$ and car $\hat{c}$ represents the case with $O_{\hat{c},J} > 1$. In this figure, t_0 is the time the algorithm performs trajectory design. x_c and $x_{\hat{c}}$ are the distance from car c and car $\hat{c}$ to the intersection,

respectively; $u_{c,J}^{opt}$ and $u_{\hat{c},J}^{opt}$ are the optimal speeds of car c and car $\hat{c}$ respectively at time t_0; and t_g is the earliest possible time when this approach can be given green signal, which is determined by the passing time of the last vehicle that has departed in the conflicting approach. Here a triangular fundamental diagram is assumed, as is shown in Fig. 3a, with free flow speed u_f, backward wave speed w and jam density k_{jam}.

3.3.1. Platoon leader

For car c which is the first car in the platoon, the optimal trajectory would let it pass the intersection exactly at time t_g and then accelerate. This assumption which is made for simplicity purposes, could cause a small and systematic error, as in theory, the car could try to accelerate before and enter the intersection with the free flow speed. This error would affect the model in a conservative way: the observed delay in simulation is probably slightly larger than what could be realized.

The optimal speed of car c before accelerating can be found by Eq. (8).

$$u_{c,J}^{opt} = \frac{x_c - \sigma_x}{t_g - t_0} \tag{8}$$

where σ_x is the estimated standard deviation for location errors. σ_x is estimated with the aid of a Kalman filter. The detailed explanation of the Kalman filter and σ_x can be found in Appendix B. Eq. (8) designs a conservative speed when there are measurement errors. In cases without measurement errors, the Kalman filter would give a $\sigma_x = 0$.

The optimal speed may not satisfy the constraint of Eq. (7) and hence should be further adjusted.

$$u_{c,J}^{des} = \begin{cases} \max\left\{\min\{u_{c,J}^{opt}; u_f\}; u_{\min}\right\}, & t_g > t_0 \\ u_f, & t_g \leqslant t_0 \end{cases} \tag{9}$$

In case $t_g > t_0$, the speed is constrained by the lower speed bound and the free flow speed. In case $t_g \leqslant t_0$, car c arrives at the intersection after the signal switches, thus car c can drive with the free flow speed on the approach link.

Then, the initial speed at which car c enters the intersection is calculated as

$$u_{c,J}^{init} = \begin{cases} u_{c,J}^{des}, & u_{c,J}^{des} > u_{\min} \\ 0, & u_{c,J}^{des} = u_{\min} \end{cases} \tag{10}$$

where in the second case ($u_{c,J}^{des} = u_{\min}$), the car would arrive at the intersection before the signal turns green. This means that the car would simply stop to avoid crawling and then accelerate across the intersection. Therefore the initial speed is 0 in this case.

3.3.2. $\hat{c}$ Platoon follower

For car $\hat{c}$ that follows the previous cars, the optimal trajectory should be tangent to the closest possible trajectory (according to Newell's car following model (Newell, 2002); see the dash line in Fig. 3b) to the last vehicle in the platoon. This means that car $\hat{c}$ will join the platoon and then accelerate across the intersection. The following equation is derived by basic kinematic law and traffic flow theory.

$$t_g + \frac{u_{\hat{c},J}^{opt} - u_{c,J}^{des}}{a} + \frac{(O_{\hat{c},J} - 1)/k_{\text{jam}}}{w} = t_0 + \frac{x_{\hat{c}}}{u_{\hat{c},J}^{opt}} - \frac{(O_{\hat{c},J} - 1)/k_{\text{jam}} + \frac{\left(u_{\hat{c},J}^{opt}\right)^2 - \left(u_{c,J}^{des}\right)^2}{2a}}{u_{\hat{c},J}^{opt}} \tag{11}$$

where both sides of the equation represent the time when the actual trajectory of car $\hat{c}$ intersects with the closest possible trajectory (marked as point of tangency in Fig. 3b). The LHS corresponds to this intersecting time in the closest possible trajectory while the RHS corresponds to this intersecting time in the designed trajectory.

The optimal speed $u_{\hat{c},k}^{opt}$ can then be obtained by solving Eq. (11).

$$u_{\hat{c},J}^{opt} = u_{c,J}^{des} - a\left(t_g - t_0 + \frac{O_{\hat{c},J} - 1}{wk_{\text{jam}}}\right) + a\sqrt{\left(t_g - t_0 + \frac{O_{\hat{c},J} - 1}{wk_{\text{jam}}}\right)^2 + 2\frac{1}{a}\left(x_{\hat{c}} - \frac{(O_{\hat{c},J} - 1)(u_{c,J}^{des} + w)}{wk_{\text{jam}}}\right)} \tag{12}$$

The design speed $u_{\hat{c},J}^{des}$ is obtained by adjusting $u_{\hat{c},J}^{opt}$ according to constraint Eq. (7).

$$u_{\hat{c},J}^{des} = \max\left\{\min\left\{u_{\hat{c},J}^{opt}; u_f\right\}; u_{\min}\right\} \tag{13}$$

The initial speed when car $\hat{c}$ enters the intersection can be found as

$$u_{\hat{c},J}^{init} = \max\left\{u_{\hat{c},J}^{des}; \sqrt{\left(u_{\hat{c},J}^{des}\right)^2 + 2a(O_{\hat{c},J} - 1)/k_{\text{jam}}}\right\} \tag{14}$$

where the first term of the RHS represents the case where this car joins the accelerating platoon after entering the intersection and the second term represents the case where this car enters the intersection with the accelerating platoon (as in Fig. 3b).

3.4. A branch and bound algorithm as a solution method

In this section, a branch and bound algorithm is proposed to search for the exact optimal departure sequence and the optimal speed for the proposed bi-level optimization model. The solution procedure can be modelled as a tree search problem, as is shown in Fig. 4.

Each node in this tree consists of a vehicle, which can be either a connected/automated vehicle, or an estimated conventional vehicle. Each node has the following attributes shown in Table 2.

The root node of this tree consists of the last vehicle that has departed from the intersection, which gives the initial state (platoon information, departure time of the previous vehicle, etc.). Each child of node n consists of the next vehicle that departs after the vehicle in node n. For this problem, each node n has at most two children (the nodes that are successive to node n), as there are two approaches to discharge from. Therefore, each path from the root node to a leaf node represents a departure sequence. Then finding the optimal departure sequence is equivalent to finding the path that minimizes the total vehicle delay.

A branch and bound algorithm is used for the tree search problem. This algorithm is based on the depth-first search, which traverses the nodes of the tree from the root node and searches as far as possible along each path before backtracking. The implementation of the depth-first search algorithm uses a stack S. When visiting each node, the algorithm calculates the departure time of the vehicle in this node (Eqs. (5) and (6)) and the optimal speed if this vehicle is automated (Eqs. (8)–(14)). A trimming mechanism is introduced based on a lower and upper estimated bound for the optimal solution. For this problem, a global upper bound, UB, of the optimal solution is defined as the minimum total delay of all the paths found so far. For each node n, a local lower bound, LB_n, is also defined such that LB_n is smaller than the total delay of any path including node n, which will be determined later in this section. Before checking the children of each node, the node is checked against these bounds, and is discarded if it cannot produce a better solution than the best one found so far. Specifically, if $LB_n > UB$, then node n, together with all of its children and descendants, should be trimmed. By applying this trimming mechanism, the computational complexity is drastically reduced.

The solution procedure is written in pseudo code and described in Algorithm 1.

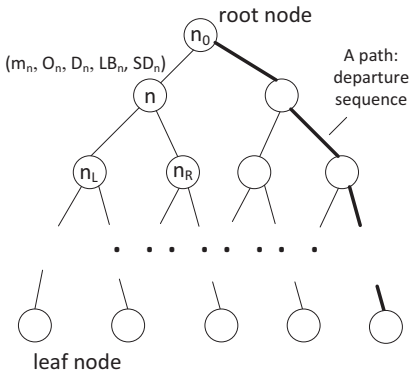


Fig. 4. Illustration of the branch and bound algorithm.

Table 2
Attributes of a node n.

m_n	Arriving approach of the vehicle in node n
O_n	Position in the platoon of the vehicle in node n
D_n	Departure time of the vehicle in node n
SD_n	Sub delay from the root node to the current node n
LB_n	Lower bound of the total delay among all the departure sequences that consist node n

Algorithm 1. Branch and bound algorithm for solving the optimal departure sequence and trajectory.

Input: vehicle sets N_i from approach $i = 1, 2$, including the speed and location information; the approach m_0, platoon O_0, and departure time D_0 of the last vehicle that has departed.
Output: the optimal departure sequence $\widehat{J}$; the optimal total delay TD; and the optimal speed $u_{c,\widehat{J}}^{opt}$ for all c in $\widehat{J}$.
1: Build the root node n_0 using m_0, O_0 and D_0
2: Calculate the lower bound LB_0 for the root node n_0 using Eq. (20)
3: For the same departure sequence J_0, calculate the global upper bound UB and u_{c,J_0}^{opt} using Eqs. (5), (6), (8)–(14).
4: Initialize the stack S and S.push(n_0)
5: Total delay $TD \leftarrow UB$
6: **while** S is not empty **do**
7: Node n = S.pop();
8: **if** $LB_n > UB$ **then**
9: Continue
10: **if** $TD > UB$ **then**
11: $TD \leftarrow UB$
12: Update the optimal departure sequence the optimal speed
13: Build the child nodes n_L and n_R of node n with the first vehicles in $N_i, i = 1, 2$.
14: Calculate the departure time D_L and D_R using Eqs. (5) and (6)
15: Calculate the sub delay from the root node to n_L and n_R, respectively, i.e. $SD_L \leftarrow SD_n + D_L$ and $SD_R \leftarrow SD_n + D_R$
16: Calculate the lower bound LB_L and LB_R, respectively, using Eq. (20)
17: S.push(n_L) and S.push(n_R)

The rest of this section finds the local lower bound LB_n for node n.

Note that from Eqs. (5) and (6), it holds that

$$D_{c,J} \geqslant D_{c',J} + \frac{1}{S_m} + \frac{l}{v_f} \tag{15}$$

where c is any vehicle and c' is the previous vehicle that departs exactly before c.

Then for any departure sequence J, define $D'_{c,J}$ such that

$$D'_{c_n,J} = D_{c_n,J} \tag{16}$$

$$D'_{c,J} = D'_{c',J} + \frac{1}{S_m} + \frac{l}{v_f} \tag{17}$$

where c_n is the vehicle in node n; c and c' vehicles in later position than c_n in departure sequence J; c' is the previous vehicle to vehicle c.

It is derived from Eqs. (15)–(17) that

$$D'_{c,J} \leqslant D_{c,J} \tag{18}$$

Denote J_n as the ordered set of vehicles in J starting from c_n. Denote SD_n as the sub delay from the root node to node n. Then it holds by Eq. (18) that

$$SD_n + \min_{J_n} \sum_{c \in J_n/\{c_n\}} D'_{c,J} \leqslant \min_J TD_J \tag{19}$$

By Eq. (19), the lower bound LB_n can be defined as

$$LB_n = SD_n + \min_{J_n} \sum_{c \in J_n/\{c_n\}} D'_{c_n,J} \tag{20}$$

Then the following model needs to be solved.

$$\min_{J_n} \sum_{c \in J_n/\{c_n\}} D'_{c_n,J} \tag{21}$$

Notice that only the first term of the penalty in Eq. (6) is considered when determining the lower bound. Then, only two possible departure sequences in vehicle set $J_n/\{c_n\}$ solves the optimization model Eq. (21) and yields the lower bound LB_n in Eq. (20): (i) the departure sequence that discharges all vehicles from approach 1 first, then discharges all vehicles from approach 2; (ii) the departure sequence that discharges all vehicles from approach 2 first and then discharges all vehicles from approach 1. These two cases are compared to find the value of LB_n using Eq. (20). Note that this lower bound may not be realized as only the first term of the penalty function Eq. (6) is considered.

4. Simulation results and algorithm evaluation

A micro-simulation platform is coded in Java to evaluate the proposed algorithm. There are two interacting layers in the simulation framework: (1) the real layer simulates the traffic dynamics using the arrival information and the control policy; (2) the control layer calculates the control policy (departure sequence and trajectory) using the real traffic information.

The real layer consists of the car following behaviors, the vehicle dynamics and the configurations of the intersection and the signal. Many car following models have been proposed over the years, either deterministic (Treiber et al., 2000; Wiedemann, 1974) or stochastic (Chen et al., 2010; Treiber et al., 2006; Menendez, 2006). For simplicity, the car following behavior in this paper is assumed as the Intelligent Driver Model (IDM) (Treiber et al., 2000) to represent a more realistic driving behavior. The parameters in IDM are calibrated using the Lankershim Dataset from the Next Generation Simulation (NGSim) project (Alexiadis et al., 2004). The maximum acceleration rate is 1.8 m/s^2, the desired deceleration rate 3 m/s^2, the minimum spacing is 2.4 m, the car length 4.8 m, the reaction time 1.4 s and the desired speed 60 km/hr. The dynamics of the cars follows the basic kinematic equations discretized by the time step 0.01 s. The length of intersection is 5 m, the maximum green time 60 s, and the minimum green time 5 s. The minimum and maximum green times are set at the same levels for both the actuated control and the proposed control algorithm.

In the control layer, the algorithm parameters are defined as: the acceleration rate $a = 1.8$ m/s^2; the saturation flow rate $S_1 = S_2 = 1800$ veh/h; free flow speed $u_f = 60$ km/h; backward wave speed $w = 20$ km/h; length of the zone of interest[3] $d = 100$ m; minimum speed for trajectory design $u_{\min} = 10$ km/h.

Notice that the car following model used in the simulation and the control algorithm is different. This is to show that the simplification in the control algorithm performs well in practice.

The total input flow (combined flow of the two approaches) is set to vary between 1000 and 2000 veh/h. Notice that the cases with a total flow of 2000 veh/hr represent oversaturated conditions. The demand ratio (ratio of total demand between the two approaches, i.e. flow on approach 1 divided by flow on approach 2) varies between 0.2 and 1. A small demand ratio means that the demand is unbalanced. Arrivals are generated randomly assuming an exponential headway distribution. The expected headway equals the inverse of the flow for a given approach.

Note that at least a small fraction of connected and/or automated vehicles are required for the operations of the algorithm as they constitute the only information source (as there are no fixed detectors). Therefore, the information level is set to vary between 0.2 and 1 in the simulation. The automated level ranges from 0 to 1. As the automated level decreases to 0, the algorithm will be reduced to the algorithm in Guler et al. (2014) (i.e. without trajectory design).

A simulation of 400 cars is run for 20 different random seeds for each scenario tested. Average delay and average number of stops per car are recorded.

The rest of this section shows the results and evaluates the algorithm. In Section 4.1, the performance of the algorithm is evaluated by comparing its performance to that of an actuated signal control algorithm. A sensitivity analysis is then conducted in Section 4.2 to evaluate the robustness of the algorithm to measurement errors. Section 4.3 analysis the robustness of this algorithm to different arrival patterns. Section 4.4 evaluates the computational efficiency of the proposed branch and bound algorithm.

4.1. Performance of the algorithm

The performance of the proposed algorithm is evaluated by comparing the resulting average car delay and average number of stops to an actuated signal control algorithm. The actuated signal control algorithm assumes the presence of fixed detectors near the intersection on both approaches. It operates as such: the signal switches to red if (1) no cars have arrived on the current approach for 5 s or (2) the green time exceeds the maximum green time of 60 s. The location of the loop detectors is chosen at 65 m upstream the intersection, which is optimized by a sensitivity analysis. Simulations are conducted for scenarios of different total flows, demand ratios, information levels and automated levels. The results for average number of stops and average delay per car are shown in Figs. 5 and 6 respectively.

The results show that the connected vehicle algorithm performs better for both average number of stops and average delay for a moderate information level. Generally speaking, this algorithm requires 50% penetration rate of vehicles that provide information to perform better than the actuated signal control algorithm. This is an expected result because the actuated signal control algorithm uses information extracted from infrastructure devices (e.g. loop detectors, video cameras, etc.), whereas the connected vehicle algorithm does not utilize any of such information. In the case where the connected vehicle algorithm could also use information from other sources (besides the connected and the automated vehicles themselves), it is expected to outperform the actuated signal control algorithm for a much lower penetration rate.

The effects of trajectory design can also be observed from the trend (i.e. monotonicity) of each curve in Figs. 5 and 6. This is because the scenarios with automated level of 0 represent the cases without trajectory design. As is shown in Figs. 5 and 6a-b, the average number of stops in all scenarios tested and the average delay in the low demand scenarios decrease

[3] A length of 100 m is chosen because this is the minimum of the typical city block size (100 m) and the average DSRC range used in the literature (300 m). Also note that this choice is conservative. The performance is expected to improve by increasing this length, as there is more information provided by connected/automated vehicles.

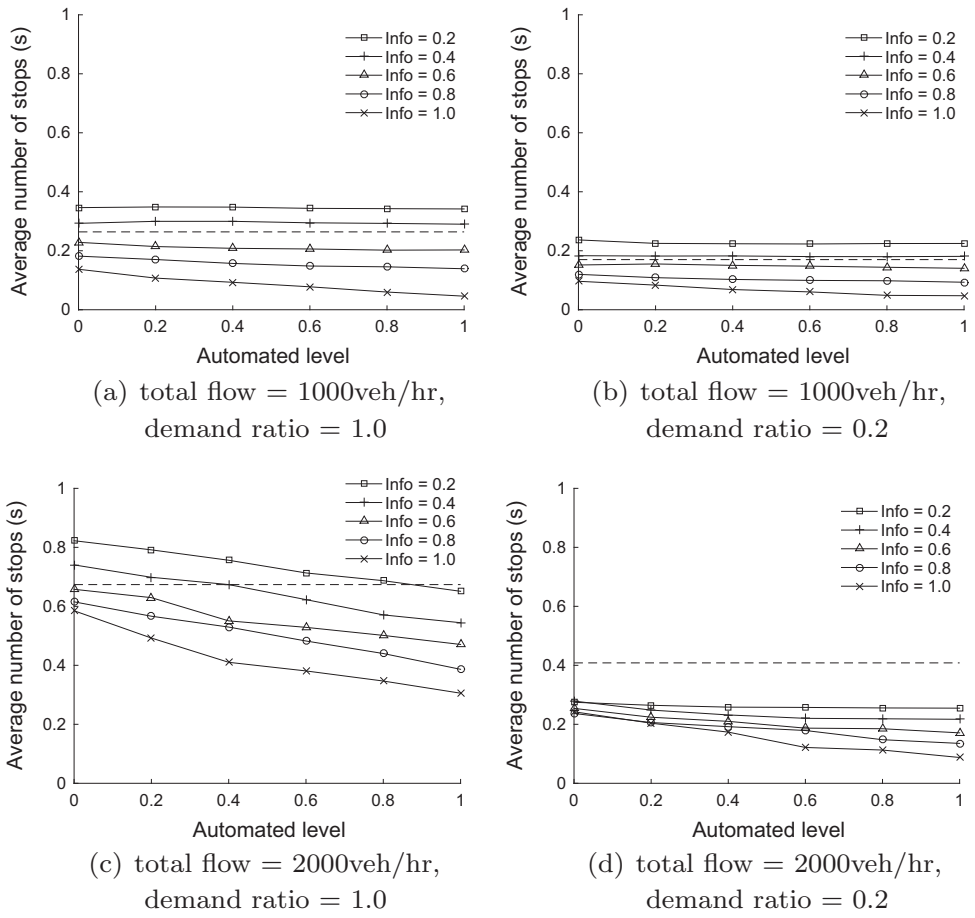


Fig. 5. Simulation results for average number of stops per car. Solid lines with different labels represent the results of the connected vehicle algorithm under different information levels. Dashed lines represent the results of the actuated signal control algorithm.

monotonously with the increase of automated level. This means that the trajectory design successfully reduces the average number of stops in all scenarios and the average delay in low demand scenarios. However, the trajectory design does not perform as well for the high demand scenarios for the average delay. In fact, for high demand scenarios, the performance of trajectory design is highly sensitive to the information level. For low information levels, the average delay might increase with the increasing percentage of automated vehicles, as is shown in Fig. 6c and d. This happens because in the high demand scenarios, there are vehicles lingering for more than one cycle and forming long queues due to oversaturation. As information drops, the algorithm will have insufficient knowledge of the arrivals and the queues. In such cases, although trajectory design is still possible, it is very likely to give an unsatisfactory solution which may give rise to a non-optimal signal timing strategy. Hence, this unsatisfactory trajectory could be detrimental to the system. For high information levels, however, the trajectory design efficiently reduces both number of stops and delay. Therefore a better approach is to use the algorithm without trajectory design in the high demand and low information cases.

4.2. Robustness to location errors

Due to the noises in measurement and the data corruption in transmission, the location information received by the central controller may not be accurate. This section tests how the errors in location affects the performance of the proposed algorithm. It is assumed that the location errors and the speed errors follow a joint Gaussian distribution. The errors (location and speed) at different time steps are assumed to be independent. The mean of the Gaussian distribution is assumed as 0 for both location and speed, which represents that there are no systematic errors. The standard deviation of the location measurement is assumed as 15.0 m and the standard deviation of the speed measurement is assumed to be 2 m/s. This is because the normal GPS devices provide an accuracy of 7.8 m at 95% confidence level (Florin and Olariu, 2015), and the accuracy can also be enhanced by map-matching (Greenfeld, 2002) or data fusion with other in-vehicle sensors (Caron et al., 2006). A Kalman filter is implemented to reduce the noise in the location measurements (a detailed description of this

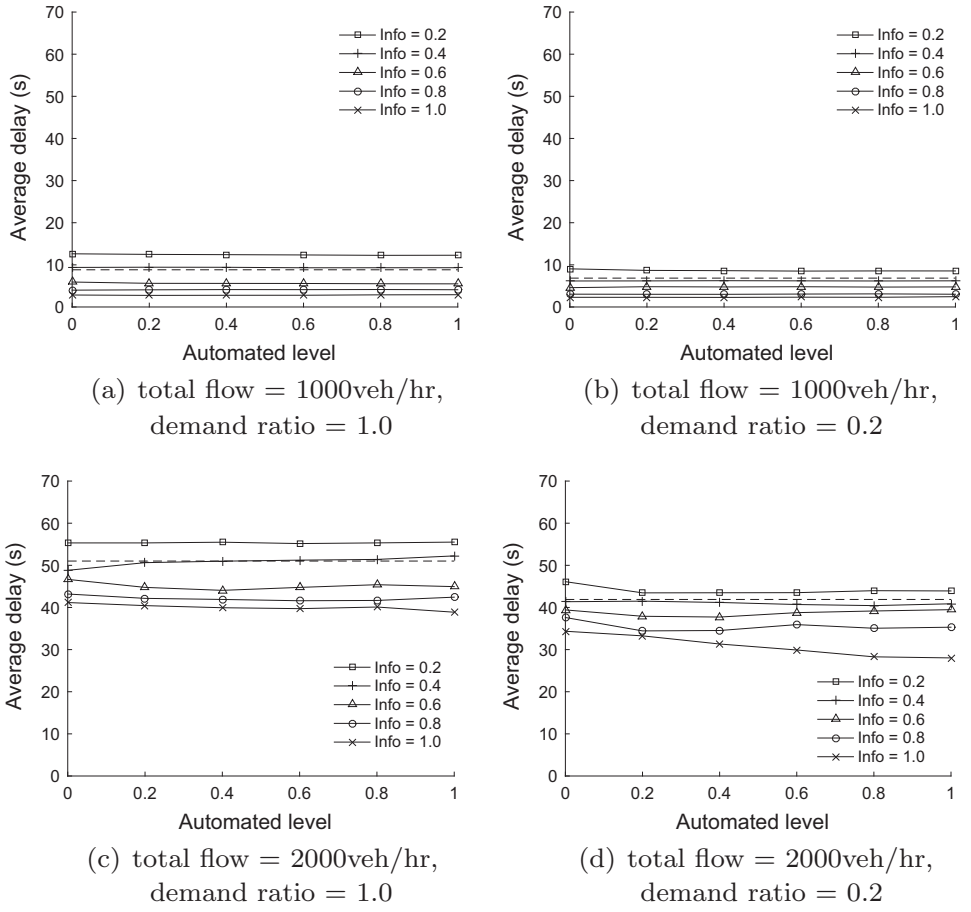


Fig. 6. Simulation results for average delay per car. Solid lines with different labels represent the results of the connected vehicle algorithm under different information levels. Dashed lines represent the results of the actuated signal control algorithm.

algorithm can be found in Appendix B). The estimation error of the location is not sensitive to the speed measurement errors after the implementation of the Kalman filter (see Fig. B.12). Then the covariance matrix is assumed to be:

$$\boldsymbol{R} = \begin{bmatrix} 225 & 0 \\ 0 & 4 \end{bmatrix} \tag{22}$$

The other simulation settings are the same as in Section 4.1. The results are shown in Fig. 7.

Fig. 7 shows that the proposed algorithm outperforms the actuated algorithm in the cases with full information even when there are measurement errors. It can also be seen that the effect of these measurement errors is larger if the information level and automation level increase. This is because as more vehicles report information and receive trajectory guidance, the system receives more noisy data.

Moreover, the actuated algorithm assumes that the loop detector data is available. If the loop detector data is integrated in the proposed algorithm, the proposed algorithm can be expected to always outperform the actuated algorithm. Also note that these results are obtained using very conservative assumptions as automated vehicles in reality cannot have large measurement noises. Therefore, this algorithm is expected to work properly in reality.

4.3. Robustness to arrival patterns

This subsection tests how robust the algorithm is to different arrival patterns. For each combination of total flow rate, demand ratio, information level and automated level, the measure for robustness is defined as the coefficient of variation, i.e. the ratio between the standard deviation and the mean of the average delay or the average number of stops. The coefficient of variation measures the dispersion of the results in the data set, and is comparable across different data sets even though they do not have the same mean.

Multiple tests show that the coefficient of variation is not sensitive to the information level or the automated level (as is shown in Fig. 8b-d), but highly related to total flow. Particularly, the coefficient of variation increases drastically with total

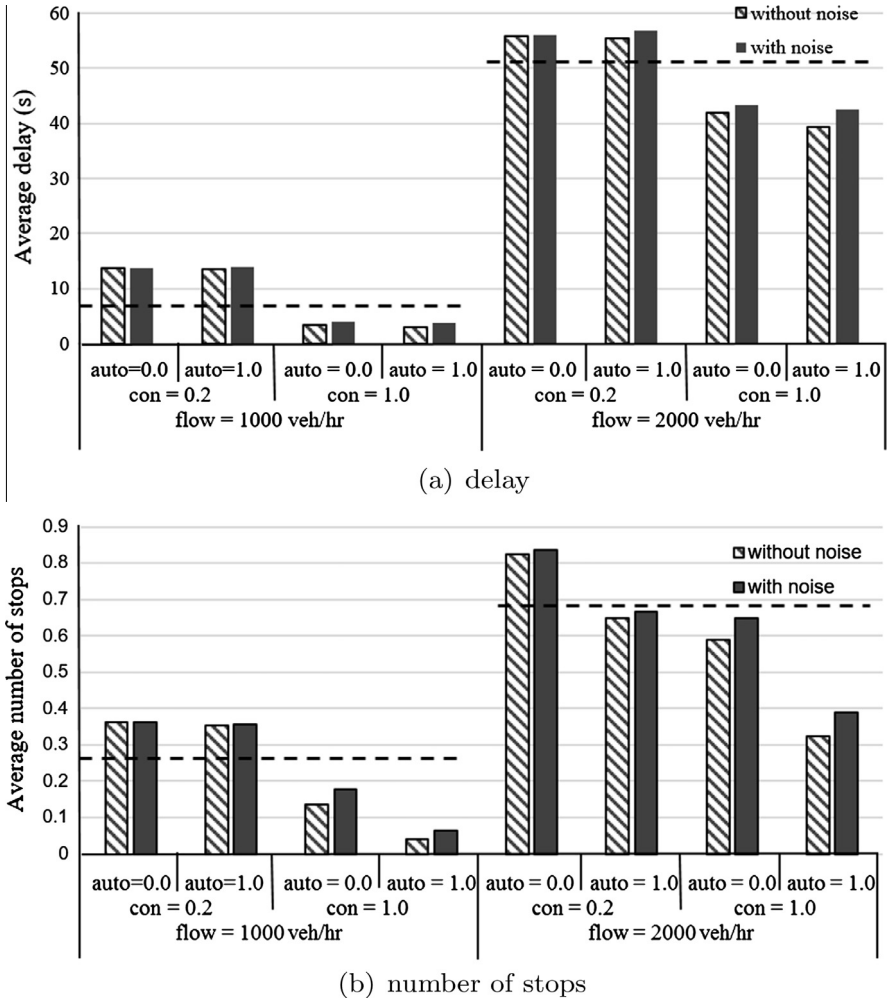


Fig. 7. Robustness to measurement errors. The dashed lines represent the results of the actuated control algorithm.

flow. The trend is very similar for all demand ratios. Therefore, for demonstration purposes, the coefficient of variation is averaged over different demand ratios, information levels and automated levels to show the impacts of total flow on the robustness. Fig. 8a shows the results.

As is shown in Fig. 8a, the coefficient of variance increases as the total flow increases. This means that the algorithm is more sensitive to the arrival patterns as the total flow increases, and the intersection becomes more saturated. Hence, the arrival patterns will have a large influence on the temporal and spatial extent of the queue, which impacts the performance of the algorithm (especially for low information levels). This also helps explain the chaotic pattern of the performance index in high demand cases in Fig. 6.

4.4. Efficiency of the branch and bound algorithm

This section tests the efficiency of the proposed branch and bound algorithm. A total of 1896 cases are tested. The algorithm finds the optimal solution in all cases tested. The efficiency of this branch and bound algorithm is evaluated by comparing the average number of nodes visited to obtain the optimum departure sequence to an enumeration algorithm. The enumeration algorithm visits all the nodes in the tree. The results are shown in Table 3.

Table 3 shows the trend of both the enumeration method and the branch and bound method when the number of cars in the zone of interest increases. It can be seen from Table 3 that the branch and bound method is significantly more efficient than the enumeration method. In contrast to the exponentially increasing complexity of the enumeration method, the computational complexity of the branch and bound increases more linearly with increasing the number of cars.

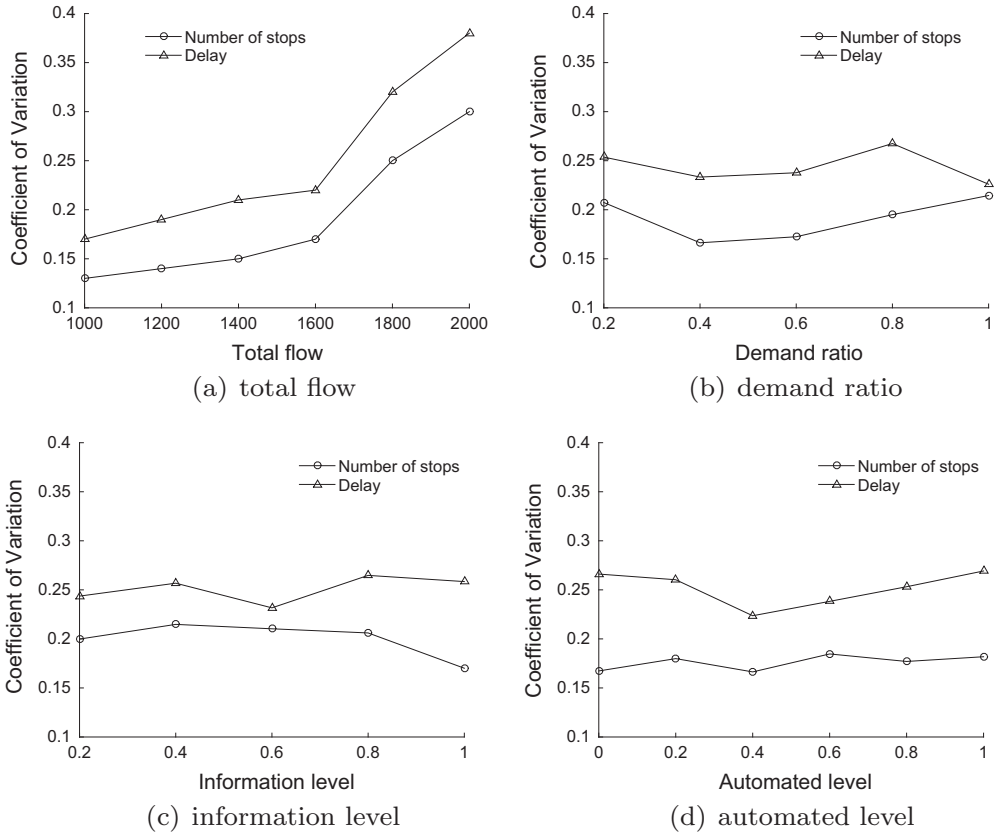


Fig. 8. Average coefficient of variation with different total flows, demand ratios, information levels and automated levels.

Table 3
Efficiency of the branch and bound algorithm (nodes visited).

# of cars in zone of interest	14	15	16	17	18	19	20	21
Enumeration	28,001	54,931	123,112	174,284	223,887	290,485	437,376	1,139,544
Branch and bound	1385	1616	1832	2236	2306	2096	2088	2404
Ratio	40.4	68.0	134.4	155.9	194.2	277.2	418.9	948.0

The average run time for each case using the branch and bound algorithm is 0.6 ms, in contrast to 26 ms using the enumeration method. The speed at which the solution can be obtained with the branch and bound algorithm is promising for increasing the complexity of the problem to account for multiple lanes and approaches.

5. A demand responsive control strategy

As discussed in Section 4, the proposed algorithm has limitations especially for low information availability cases. In this section, a demand responsive control strategy is proposed to adapt to different traffic situations. This strategy dynamically switches between the two (or three) algorithms based on demand and information level: the connected vehicle algorithm with and without trajectory design, and the actuated signal control algorithm if the necessary infrastructure (e.g. detectors) is available.

The demand responsive strategy is determined by identifying the bounds for each algorithm. To do so, simulation is conducted for different total flows, demand ratios, information levels and automated levels. Other parameters remain the same. Simulation results are shown in Fig. 9.[4] Two polylines represent the boundaries for the three algorithms. Notice that those boundaries are conservative because loop detector information is not integrated into the connected vehicle algorithm, but is used for actuated signal control.

[4] The upper boundary determines when the trajectory design should be used. If the system is worse off with the trajectory design, then the algorithm without trajectory design is advised by Fig. 7.

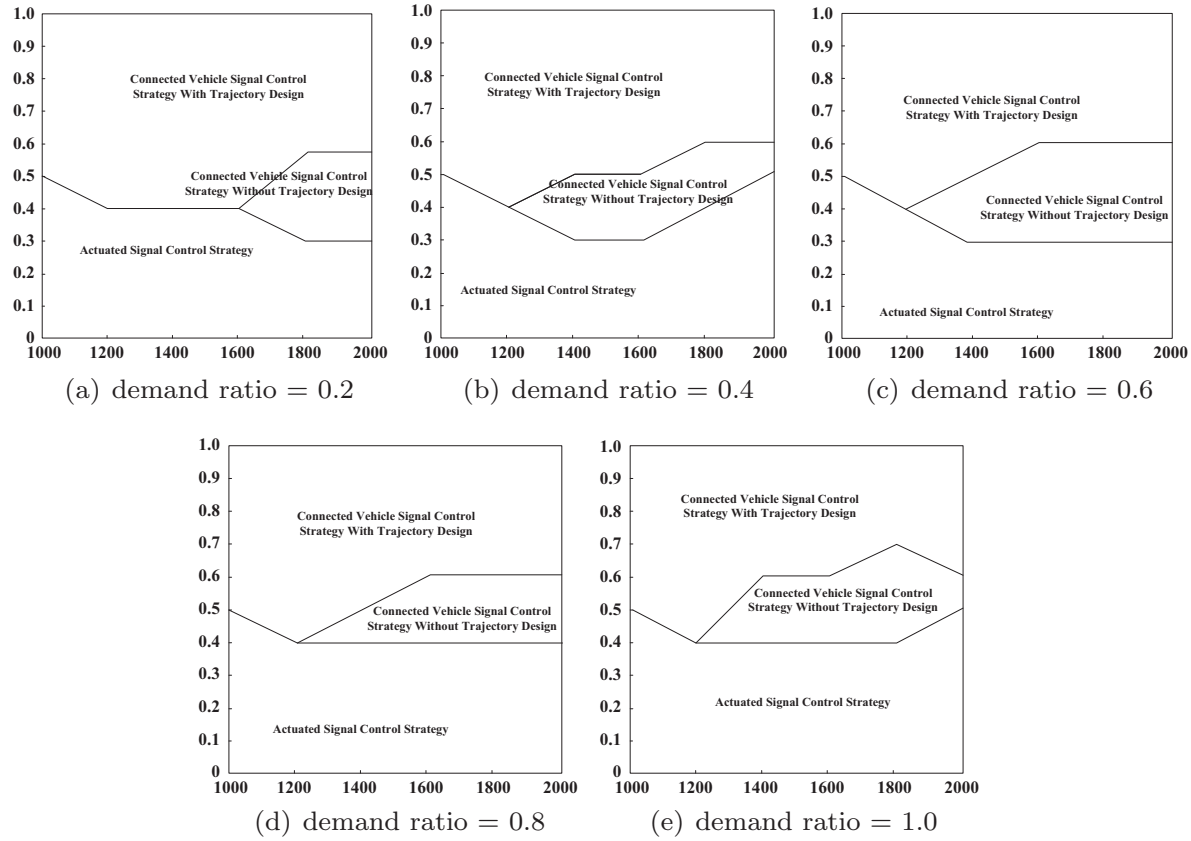


Fig. 9. Demand responsive signal control strategy based on connected vehicle technology. The horizontal axes represent total flow (veh/hr); the vertical axes represent information level.

The upper boundary distinguishes between the connected vehicle strategies with and without trajectory design. The boundary is calculated as the minimum information level such that using trajectory design gives a better result in both average delay and average number of stops. In other words, for a certain combination of information level and total flow, trajectory design is only adopted if both average delay and average number of stops can be improved for all automated levels ($\geqslant 0.2$). This is a conservative application bound.

The lower boundary, i.e. the boundary between actuated signal control strategy and connected vehicle signal control strategy, is determined by the minimum information level such that connected vehicle strategy performs better than the actuated signal control strategy in terms of both average delay and average number of stops. For a given information level, demand ratio and total flow, the connected vehicle strategy is chosen only if both performance indices are improved for all automated levels. Similar to the upper one, this is also a conservative application bound.

The two boundaries divide the traffic parameter space into three parts, each of which corresponds to one of the three algorithms considered. When the current traffic situation lies in a certain part of this figure, the algorithm corresponding to that zone will be used. It is shown in Fig. 9 that the connected vehicle algorithm always requires less than 50% vehicles to provide information in order to outperform the actuated signal control algorithm. This is less strict than most of the existing research works (He et al., 2012; Z. Li et al., 2014) that require all or a majority of the vehicles to be connected or automated.

The application bounds can be relaxed if loop detector information is integrated into the connected vehicle algorithm. Simulations show that the performance is sensitive to the location of the loop detector (Yang et al., 2015). If information from a loop detector installed further than 30 m upstream of the intersection is used, the connected vehicle algorithm can always achieve a better performance than the actuated signal control algorithm. Hence, only two strategies should be considered (namely, the connected vehicle algorithm with and without trajectory design).

6. Conclusions

This paper improves the algorithm from Guler et al. (2014) by (1) integrating three different stages of technology development; (2) developing a heuristics to switch the signal controls depending on the stage of technology; (3) allowing

trajectory modification for automated cars for a simple and single intersection; (4) developing a branch and bound algorithm to substitute the enumeration method to improve the computational efficiency; (5) adopting a Kalman filter to reduce the impact of the measurement errors.

With such modifications, both car delays and number of stops are reduced compared to Guler et al. (2014) in most scenarios tested. Various simulations are conducted to evaluate the performance of the proposed algorithm. The results show that with information levels greater than 50%, the delay and number of stops can be reduced compared to the actuated signal control algorithm. The benefit of trajectory modification is greater in low demand scenarios or high demand scenarios with high information levels. A demand responsive strategy that combines the proposed algorithm with and without trajectory design and a simple actuated signal control algorithm was proposed based on the simulation results. It is also shown that the algorithm is not sensitive to measurement noises.

The arrival sequence and trajectories of conventional vehicles are estimated using kinematic wave theory and Newell's car following model. The estimation accuracy largely depends on the information level, as connected vehicles and automated vehicles are the only source of information. For low information levels, the inaccurate estimation of conventional vehicles might make the algorithm less efficient. The performance of this algorithm can be further improved if (1) information from other sources (e.g. loop detectors) is integrated (in such case a data fusion algorithm similar to that in Ambühl and Menendez, 2016, but at a microscopic level could be implemented), (2) information on the conventional vehicles is better estimated by more advanced queue estimation methods (e.g., as shown in Ramezani and Geroliminis, 2014; Comert, 2013). The performance of this algorithm can also be improved using stochastic programming (e.g., Tong et al., 2015), if the distribution of queue length is known.

This algorithm uses Newell's car following model to simplify the calculation of the trajectory design. In principle, any car following model can be used here. As long as the trajectories of previous cars are given, the design speed to enter the intersection can be determined.

The results of this paper can also be generalized to show the influence of package loss (i.e. the loss of information when a vehicle communicates to the central controller). Such loss is equivalent to reducing information level or automated level. As discussed in Section 4, the performance indices would be worse if information level is lower. However, for package losses within a certain range, there would not be much deterioration in performance.

Future work includes further generalization of the algorithm to a complex intersection. The increase in complexity can be driven by two factors: more modes or more intersection movements. In multi-modal scenarios, there might be some traffic modes requesting priority, e.g. transit vehicles, emergency vehicles, etc. Additionally, the priority scheme could follow different approaches (e.g., Guler and Menendez, 2014; He et al., 2016; Guler et al., 2016) One general way to deal with the priority requests would be to change the objective function, e.g. total passenger delay (transit vehicles), total value of time (emergency vehicles), etc. On the other hand, the proposed fast branch and bound algorithm makes it possible to generalize the model for more intersection movements. It is expected that for a larger scale of intersection, the model can be solved with reasonable computational time. A difficulty aroused from considering a more complex intersection is, however, the planning of the turning trajectories and lane changings with the co-existent of conventional, connected and automated vehicles. Ongoing efforts are being made in this direction.

Other possible extension of this paper is arterial control, which involves the cooperation not only between vehicles and intersections, but also among intersections. This is also ongoing work of the authors.

Appendix A. Estimation of the arrival information of conventional vehicles

In this section, the arrival information (arrival sequence, virtual departure time and trajectories) of conventional vehicles is estimated based on kinematic wave theory and Newell's car following model. The estimation of the arrival information of conventional vehicles is based on the information given by stopped connected or automated vehicles. There are four steps.

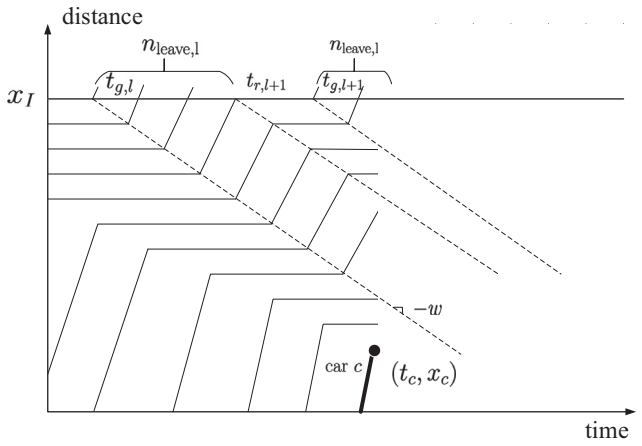
Step 1. For each approach, define the vehicle set B_l as the set of the vehicles that are forced to stop due to the lth red signal that starts at time $t_{r,l}$ and ends at time $t_{g,l}$ ($t_{r,l} < t_{g,l}$). Denote x_I as the location of the intersection.
When a connected or automated vehicle c comes to a stop at time t_c and location x_c, attach the trajectory point (t_c, x_c) to cycles, i.e. find the cycle index l such that $c \in B_l$.
Note that $c \in B_l$ if and only if

$$x_I - w(t_c - t_{g,l-1}) \leqslant x_c \leqslant x_I - w(t_c - t_{g,l}) \tag{A.1}$$

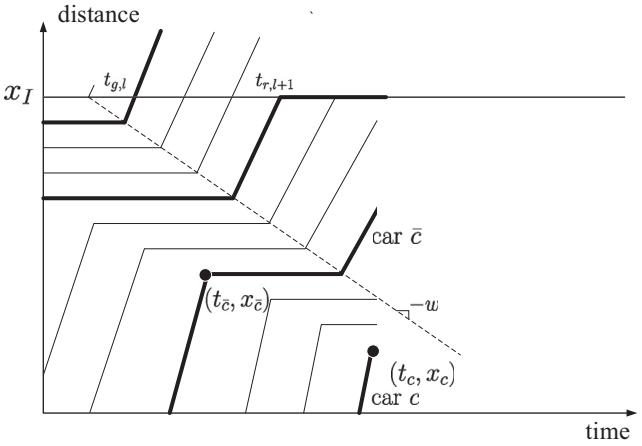
where w is the backward wave speed. Eq. (A.1) means that the trajectory point (t_c, x_c) is inside the region between the discharging lines at the $(l-1)$th green time and lth green time.

Step 2. For B_l that contains car c, sort the vehicles in ascending order according to arrival times of the vehicles.

Step 3. Calculate the arrival sequence of conventional vehicles. There are two cases based on whether or not car c is the first connected or automated vehicle attached to its cycle B_l.
Fig. A.10a illustrates the case where car c is the first connected or automated vehicle attached to its cycle. This means that any car in front of car c is a conventional vehicle. In this case, the number of conventional vehicles n_{conv} in front of car c can be estimated as



(a) car c is the first connected/automated vehicle attached to its cycle



(b) car c is not the first connected/automated vehicle attached to its cycle

Fig. A.10. Illustration of the estimation of arrival sequence of conventional vehicles. The bold lines represent the trajectories of the connected or automated vehicles and the normal lines represent the trajectories of the conventional vehicles.

$$n_{\text{conv}} = \lfloor x_c k_j \rfloor - \lfloor T_g S_m \rfloor \tag{A.2}$$

where T_g is the sum of green time from time $t_{g,l}$ to time t_c, $\lfloor T_g S_m \rfloor$ is the number of vehicles that leave the intersection from time $t_{g,l}$ to time t_c.

Fig. A.10b represents the case where car c is not the first connected or automated car in B_l. Denote the previous connected or automated car as $\bar{c}$. Denote the time and location car $\bar{c}$ comes to a stop due to the red signal as $t_{\bar{c}}$, $x_{\bar{c}}$. Then the number of conventional vehicles between car c and car $\bar{c}$ is

$$n_{\text{conv}} = \lfloor (x_c - x_{\bar{c}}) k_j \rfloor \tag{A.3}$$

Step 4. Estimate the virtual departure time and the trajectories of the conventional vehicles. The virtual departure time of the conventional vehicles is estimated using linear interpolation between the virtual departure times of connected or automated vehicles. The estimation of trajectories of conventional vehicles is based on Newell's model.

The algorithm is validated based on NGSIM data (Lankershim trajectory data from 8:30 a.m. to 8:45 a.m. on June 16, 2005) to compare the queue length. The mean estimation error is estimated, i.e. the average absolute difference between the estimated queue length and the actual queue length. The results are shown in Fig. A.11. It can be seen that the estimation error is within 3 cars for penetration rate larger than 45%.

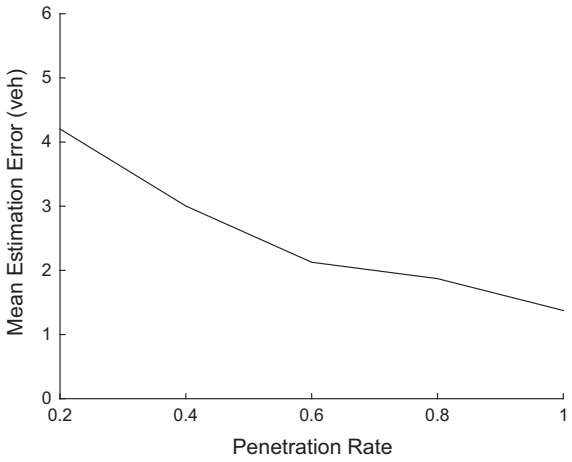


Fig. A.11. Validation result of the estimation algorithm with NGSim data.

Appendix B. Kalman filter for location and speed estimation with measurement noise

Kalman filter (Kalman, 1960) is an algorithm that produces estimates of unknown variables from a time series of noisy measurements. The estimation results are more precise than a single measurement. In this section, a Kalman filter is applied to estimate the speed x and location u, as well as the standard deviation of the location σ_x. The signal control algorithm uses this information as an input.

The Kalman filter is applied in an online manner. Every time the vehicle sends information, the estimated location $\hat{x}(k)$, speed $\hat{v}(k)$ and their covariance matrix $\mathbf{\Sigma}(k)$ are updated. In this section, all the variables with bold font represent vectors or matrices.

A Kalman filter consists of system equations and measurement equations. The system equations are derived from the basic vehicle motion equations.

$$x(k) = x(k-1) + u(k-1)C + q_x(k) \tag{B.1}$$

$$u(k) = u(k-1) + q_u(k) \tag{B.2}$$

where $x(k)$ and $u(k)$ represent the real location and speed at time step k; C is the duration of each time step, which is chosen as 0.2 s in this paper (according to Gómez et al. (2015) and Leung et al. (2006), a resolution between 1 and 5 Hz is reasonable for GPS data). $q_x(k)$ and $q_u(k)$ are the system errors at time step k, which are introduced due to two reasons: (1) acceleration and driving behavior are not considered; (2) time is discretized. The joint distribution of $q_x(k)$ and $q_u(k)$ is assumed to follow a two dimensional Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{Q}$, i.e.

$$[q_x(k), q_v(k)]^T \sim N(\mathbf{0}, \boldsymbol{Q}) \tag{B.3}$$

where x^T represents the transpose of vector/matrix x. Covariance matrix $\boldsymbol{Q}$ can be calibrated from real data, and is related to the sampling interval of the GPS data. In this paper, the following $\boldsymbol{Q}$ is used.

$$\boldsymbol{Q} = \begin{bmatrix} 6.8061 & 0.0382 \\ 0.0382 & 0.3819 \end{bmatrix} \tag{B.4}$$

The measurement equations are written as

$$y(k) = x(k) + r_x(k) \tag{B.5}$$

$$v(k) = u(k) + r_u(k) \tag{B.6}$$

where $y(k)$ and $v(k)$ are the measurement of location and speed, respectively. $r_x(k)$ and $r_u(k)$ are measurement errors of location and speed, respectively. The joint distribution of $r_x(k)$ and $r_u(k)$ is assumed to follow a two dimensional Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{R}$, i.e.

$$[r_x(k), r_v(k)]^T \sim N(\mathbf{0}, \boldsymbol{R}) \tag{B.7}$$

Covariance matrix $\boldsymbol{R}$ is related to the accuracy of the measurement devices.

To solve the Kalman filter, the system equations and measurement equations are rewritten as

$$\begin{bmatrix} x(k) \\ u(k) \end{bmatrix} = \boldsymbol{A}\begin{bmatrix} x(k-1) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} q_x(k) \\ q_u(k) \end{bmatrix} \tag{B.8}$$

$$\begin{bmatrix} y(k) \\ v(k) \end{bmatrix} = \boldsymbol{H}\begin{bmatrix} y(k-1) \\ v(k-1) \end{bmatrix} + \begin{bmatrix} r_x(k) \\ r_u(k) \end{bmatrix} \tag{B.9}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 1 & C \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{B.10}$$

Then the location, speed and their covariance can be estimated using the following iterative equations.

$$\boldsymbol{\mu}(k) = \boldsymbol{A}\begin{bmatrix} \hat{x}(k-1) \\ \hat{u}(k-1) \end{bmatrix} \tag{B.11}$$

$$\boldsymbol{P}(k) = \boldsymbol{A}\boldsymbol{\Sigma}(k-1)\boldsymbol{A}^T + \boldsymbol{Q} \tag{B.12}$$

$$\boldsymbol{K}(k) = \boldsymbol{P}(k)\boldsymbol{H}^T(\boldsymbol{H}^T\boldsymbol{P}(k)\boldsymbol{H} + \boldsymbol{R})^{-1} \tag{B.13}$$

$$\begin{bmatrix} \hat{x}(k-1) \\ \hat{u}(k-1) \end{bmatrix} = \boldsymbol{\mu}(k) + \boldsymbol{K}(k)\begin{bmatrix} y(k) \\ v(k) \end{bmatrix} \tag{B.14}$$

$$\boldsymbol{\Sigma}(k) = (\boldsymbol{I} - \boldsymbol{K}(k)\boldsymbol{H})\boldsymbol{P}(k) \tag{B.15}$$

where $\boldsymbol{\mu}(k)$ is the predicted state variables, $\boldsymbol{P}(k)$ is the predicted covariance, $\boldsymbol{K}(k)$ is the Kalman gain.

Then, by rewriting $\boldsymbol{\Sigma}(k)$ as

$$\boldsymbol{\Sigma}(k) = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xu} \\ \Sigma_{xu} & \Sigma_{uu} \end{bmatrix}, \tag{B.16}$$

the standard deviation of the location can be found as

$$\sigma_x = \sqrt{\Sigma_{xx}}. \tag{B.17}$$

The initial covariance is chosen as $\boldsymbol{R}$ and the initial speed and location are chosen as the initial measurement.

The effect of the Kalman filter can be seen in Fig. B.12. The horizontal axis is the standard deviation of the measurement errors, whereas the vertical axis represents the standard deviation of the location estimation after implementing the Kalman filter. While this paper focuses more so on the error in the location measurement since the algorithm relies more on this information, three different standard errors for speed measurements are also shown in this figure. Fig. B.12 shows that the estimation error of the location is smaller than the measurement error. In other words, by using the Kalman filter, the error related to location measurement is reduced. It is also shown in Fig. B.12 that the estimation of the Kalman filter is not very sensitive to the standard deviation of the error in speed measurement. Also notice that adding the Kalman filter would not influence the running time, as only one iteration of the Kalman equations is required in each time step.

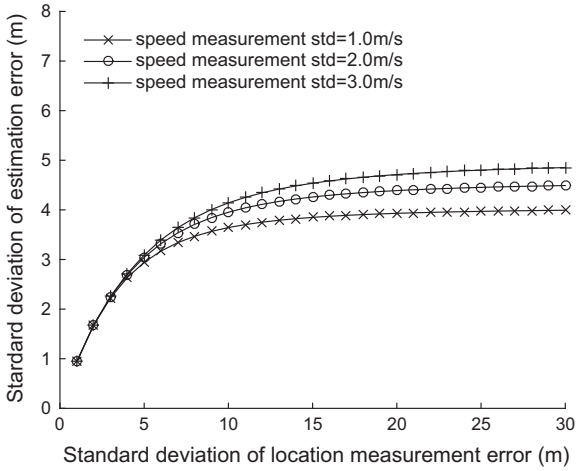


Fig. B.12. Effect of the Kalman filter.

References

Alexiadis, V., Colyar, J., Halkias, J., Hranac, R., McHale, G., 2004. The next generation simulation program. Inst. Transp. Eng. ITE J. 74, 22.

Ambühl, L., Menendez, M., 2016. Data fusion algorithm for macroscopic fundamental diagram estimation. Transp. Res. Part C: Emerg. Technol. 71, 184–197.

Andrews, S., Cops, M., 2009. Final Report: Vehicle Infrastructure Integration Proof of Concept Executive Summary–Vehicle. US DOT, IntelliDrive (sm) Report FHWA-JPO-09-003.

Au, T.C., Stone, P., 2010. Motion planning algorithms for autonomous intersection management. In: Bridging the Gap Between Task and Motion Planning.

Caron, F., Duflos, E., Pomorski, D., Vanheeghe, P., 2006. GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects. Inform. Fusion 7, 221–230.

Chen, X., Li, L., Zhang, Y., 2010. A markov model for headway/spacing distribution of road traffic. IEEE Trans. Intell. Transp. Syst. 11, 773–785.

Coelho, M.C., Farias, T.L., Rouphail, N.M., 2005. Impact of speed control traffic signals on pollutant emissions. Transp. Res. Part D: Transp. Environ. 10, 323–340.

Comert, G., 2013. Simple analytical models for estimating the queue lengths from probe vehicles at traffic signals. Transp. Res. Part B: Methodol. 55, 59–74.

Dresner, K., Stone, P., 2004. Multiagent traffic management: a reservation-based intersection control mechanism. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, vol. 2, pp. 530–537.

Dresner, K., Stone, P., 2006. Human-usable and emergency vehicle-aware control policies for autonomous intersection management. In: AAMAS 2006 Workshop on Agents in Traffic and Transportation.

Feng, Y., Head, K.L., Khoshmagham, S., Zamanipour, M., 2015. A real-time adaptive signal control in a connected vehicle environment. Transp. Res. Part C: Emerg. Technol. 55, 460–473 (Engineering and Applied Sciences Optimization (OPT-i) - Professor Matthew G. Karlaftis Memorial Issue).

Florin, R., Olariu, S., 2015. A survey of vehicular communications for traffic signal optimization. Veh. Commun. 2, 70–79.

Gómez, P., Menéndez, M., Mérida-Casermeiro, E., 2015. Evaluation of trade-offs between two data sources for the accurate estimation of origin–destination matrices. Transportmet. B: Transp. Dyn. 3, 222–245.

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

Greenfeld, J.S., 2002. Matching GPS observations to locations on a digital map. In: Transportation Research Board 81st Annual Meeting.

Guler, I., Menendez, M., 2014. Analytical formulation and empirical evaluation of pre-signals for bus priority. Transp. Res. Part B: Meth. 64, 41–53.

Guler, S.I., Menendez, M., Meier, L., 2014. Using connected vehicle technology to improve the efficiency of intersections. Transp. Res. Part C: Emerg. Technol. 46, 121–131.

Guler, I., Gayah, V., Menendez, M., 2016. Bus priority at signalized intersections with single-lane approaches: a novel pre-signal strategy. Transp. Res. Part C: Emerg. Technol. 63, 51–70.

Guo, J., Balon, N., 2006. Vehicular Ad hoc Networks and Dedicated Short-range Communication. Technical Report. University of Michigan - Dearborn.

He, Q., Head, K.L., Ding, J., 2012. Pamscod: Platoon-based arterial multi-modal signal control with online data. Transp. Res. Part C: Emerg. Technol. 20, 164–184.

He, Q., Head, K.L., Ding, J., 2014. Multi-modal traffic signal control with priority, signal actuation and coordination. Transp. Res. Part C: Emerg. Technol. 46, 65–82.

He, H., Guler, I., Menendez, M., 2016. Adaptive control algorithm to provide bus priority with a pre-signal. Transp. Res. Part C: Emerg. Technol. 64, 28–44.

Hu, J., Park, B.B., Lee, Y.J., 2015. Coordinated transit signal priority supporting transit progression under connected vehicle technology. Transp. Res. Part C: Emerg. Technol. 55, 393–408.

Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. J. Basic Eng. 82, 35–45.

Kamal, M., Imura, J.I., Hayakawa, T., Ohata, A., Aihara, K., 2015. A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights. IEEE Trans. Intell. Transp. Syst. 16, 1136–1147.

Lee, J., Park, B., 2012. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. IEEE Trans. Intell. Transp. Syst. 13, 81–90.

Lee, J., Park, B., Yun, I., 2013. Cumulative travel-time responsive real-time intersection control algorithm in the connected vehicle environment. J. Transp. Eng. 139, 1020–1029.

Leung, K.Y.K., Dao, T.S., Clark, C.M., Huissoon, J.P., 2006. Development of a microscopic traffic simulator for inter-vehicle communication application research. In: 2006 IEEE Intelligent Transportation Systems Conference. IEEE, pp. 1286–1291.

Li, L., Wang, F.Y., 2006. Cooperative driving at blind crossings using intervehicle communication. IEEE Trans. Veh. Technol. 55, 1712–1724.

Li, L., Wen, D., Yao, D., 2014. A survey of traffic control with vehicular communications. IEEE Trans. Intell. Transp. Syst. 15, 425–432.

Li, M., Boriboonsomsin, K., Wu, G., Zhang, W.B., Barth, M., 2009. Traffic energy and emission reductions at signalized intersections: a study of the benefits of advanced driver information. Int. J. Intell. Transp. Syst. Res. 7, 49–58.

Li, Z., Elefteriadou, L., Ranka, S., 2014. Signal control optimization for automated vehicles at isolated signalized intersections. Transp. Res. Part C: Emerg. Technol. 49, 1–18.

Menendez, M., 2006. An analysis of HOV lanes: their impact on traffic (PhD thesis). Department of Civil and Environmental Engineering, University of California, Berkeley, CA.

National Transportation Operations Coalition (NTOC), 2012. National Traffic Signal Report Card, Executive Summary. Nat. Transp. Oper. Coalition, Washington, DC, USA.

Newell, G.F., 2002. A simplified car-following theory: a lower order model. Transp. Res. Part B: Methodol. 36, 195–205.

Pandit, K., Ghosal, D., Zhang, H.M., Chuah, C.N., 2013. Adaptive traffic signal control with vehicular ad hoc networks. IEEE Trans. Veh. Technol. 62, 1459–1471.

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, pp. 1–6.

Ramezani, M., Geroliminis, N., 2014. Queue profile estimation in congested urban networks with probe data. Comput.-Aided Civ. Infrastruct. Eng.

Tong, Y., Zhao, L., Li, L., Zhang, Y., 2015. Stochastic programming model for oversaturated intersection signal timing. Transp. Res. Part C: Emerg. Technol. 58, 474–486.

Treiber, M., Hennecke, A., Helbing, D., 2000. Congested traffic states in empirical observations and microscopic simulations. Phys. Rev. E 62, 1805.

Treiber, M., Kesting, A., Helbing, D., 2006. Understanding widely scattered traffic flows, the capacity drop, and platoons as effects of variance-driven time gaps. Phys. Rev. E 74, 016123.

Unal, A., Rouphail, N., Frey, H., 2003. Effect of arterial signalization and level of service on measured vehicle emissions. Transp. Res. Rec.: J. Transp. Res. Board, 47–56.

Wiedemann, R., 1974. Simulation des strassenverkehrsflusses. Schriftenreihe des Instituts fr Verkehrswesen der Universitt Karlsruhe, Band 8, Karlsruhe, Germany.

Wu, J., Abbas-Turki, A., Correia, A., El Moudni, A., 2007. Discrete intersection signal control. In: IEEE International Conference on Service Operations and Logistics, and Informatics, 2007. SOLI 2007, pp. 1–6.

Xia, H., Boriboonsomsin, K., Schweizer, F., Winckler, A., Zhou, K., Zhang, W.B., Barth, M., 2012. Field operational testing of eco-approach technology at a fixed-time signalized intersection. In: 2012 15th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 188–193.

Yang, K., Guler, I., Menendez, M., 2015. A signal control strategy using connected vehicles and loop detector information. In: Proc. 15rd Swiss Transport Research Conference (Monte Verita, Switzerland).