# Adaptive Traffic Signal Control using Vehicle-to-Infrastructure Communication: A Technical Note

Chen Cai
National ICT Australia
University of New South Wales
Lv.4, 223 Anzac Parade
Kensington, NSW 2052
+61 2 8306 0421
chen.cai@nicta.com.au

Yang Wang
National ICT Australia
University of New South Wales
Lv.4, 223 Anzac Parade
Kensington, NSW 2052
+61 2 8306 0538
yang.wang@nicta.com.au

Glenn Geers
National ICT Australia
University of New South Wales
Lv.4, 223 Anzac Parade
Kensington, NSW 2052
+61 2 8306 0439
glenn.geers@nicta.com.au

## ABSTRACT
This paper presents a preliminary study on controlling traffic signals using information collected via vehicle-to-infrastructure (V2I) communication. The key idea is to use vehicle speed and position as state variable, and construct a state-space presentation of the control problem. We propose dynamic programming and its derivative methods to solve the problem. An advantage of using speed and position as state variables is that difficulties of defining queue and estimating queue length at real-time are circumvented. Dynamic programming methods aim to optimise control performance successively at real-time. The method presented here is the first few that address the increasing possibility of adopting V2I for urban traffic management.

## Categories and Subject Descriptors
G.1.6 [**Mathematics of Computing**]: Optimisation – *stochastic programming.*

## General Terms
Algorithms, Theory

## Keywords
V2I, traffic signal, computational transportation science (CTS)

## 1. BACKGROUND
Vehicle-to-Infrastructure (V2I) communication is an emerging technological framework that aims for direct data transmission between vehicles and road infrastructure using wireless communication. Data from vehicles may include speed, position, destination and occupancy; data from the infrastructure may encompass congestion, surrounding road conditions, controller status and routing information. Substantial improvement in transport efficiency using V2I technologies was seen in several pilot projects. Notable examples include Audi [1] and BMW [2] (Figure 1).


IWCTS'10, Nov 2, 2010 San Jose, CA, U.S.A.


Academic literature in this area has primarily focused on wireless communication technologies (Tarng and Chuang [3], Plainchault *et al.* [4], Korkmarz *et al.* [5], Sukuvaara and Nurmi [6], Belanovic *et al.* [7]) and improving transport safety (Misener and Shladover [8]). The implication of V2I for traffic control at intersections has yet to be addressed at a theoretical level.

With much richer and more precise information of approaching vehicles there are opportunities to develop a new generation of traffic controller that exploits the accuracy and abundance of traffic information. Instead of representing the traffic condition as degree of saturation or vehicles in a queue, which conventional adaptive controllers usually do, the new controllers will receive speed and position information (at the very least) from individual vehicles. This allows the controller to directly calculate travel time for each vehicle thus avoiding the difficulties of defining and estimating queue length in real-time.

With V2I communication, the ideal traffic controller should make successive decisions to adjust signal timings in real-time. Using the state-space representation, dynamic programming [9] is the only exact method for generating an optimal solution, but is often computationally intractable because of the issues of dimensionality. Dimensionality is especially concerning if we attempt to use individual speed and position as state variables for the control problem. Approximation technique may help to reduce computational demand, and were investigated for traffic signal control in [10] and [11]. The two studies, however, used queue length to describe the state of the system.

In this paper we present a preliminary study on a DP algorithm using vehicle speed, position and waiting time as state variables. The reward function is defined as the difference in travel time after state transition. The rest of the paper is organized as follows. We first present notations and definition in Section 2 and 3. System dynamics and the DP algorithm for traffic control using V2I are introduced in Section 4. We discuss the dimensionality issues related to the DP algorithm in Section 5. A concise

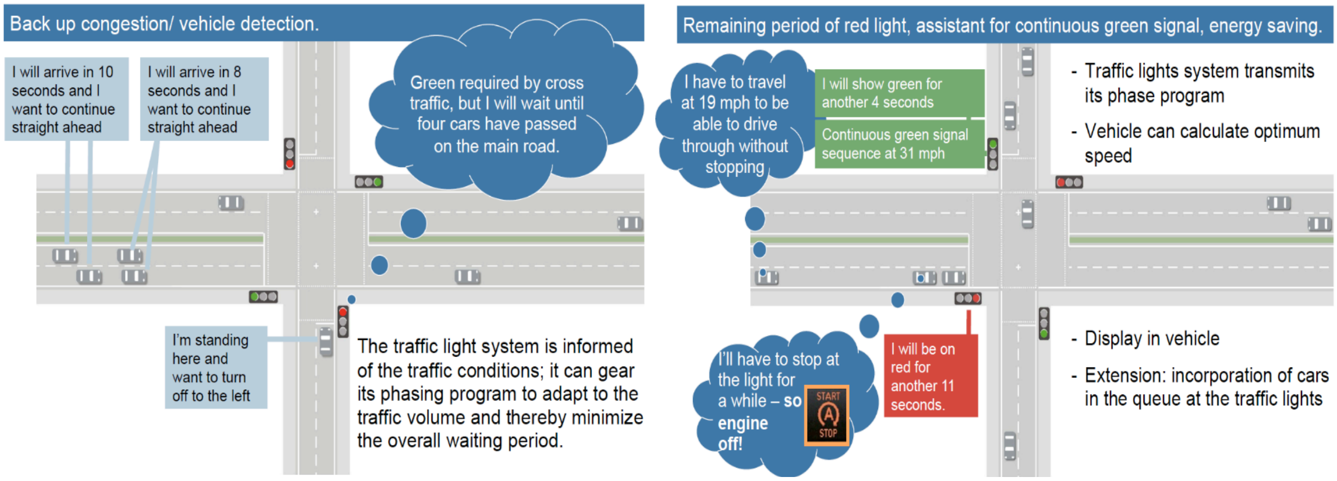


**Figure 1. Pilot experiment using vehicle-to-infrastructure communication by BMW Group. Source: Dirk Kessler, BMW Group, 2009.**

conclusion is presented in Section 6. After this, we discuss possible solutions methodologies and their associated approximations.

## 2. NOTATIONS

For ease of further discussion we define the following variables:

| | |
|---|---|
| $\mathbf{i}$ | system state, |
| $\mathbf{q}$ | traffic state |
| $\mathbf{r}$ | vector of functional parameters, |
| $\mathbf{s}$ | controller state |
| $\mathbf{u}$ | decision vector, |
| $\mathbf{z}$ | vehicle state |
| $a$ | acceleration (assumed constant), |
| $d$ | constant deceleration, |
| $l$ | position as distance to stopline, |
| $x$ | travel time |
| $v$ | vehicle speed, |
| $w$ | accmulated waiting time |
| $\alpha$ | discount factor, |
| $J(\mathbf{i})$ | the true value function associated with state $\mathbf{i}$, |
| $\tilde{J}(\cdot,r)$ | approximation to $J(\mathbf{i})$, |
| $\Delta\mathbf{r}$ | incremental adjustment to function parameters, |
| $f(\cdot)$ | ifunction that returns $\Delta\mathbf{r}$, |
| $g(\cdot)$ | one-step cost function, |

## 3. DEFINITIONS

We assume that V2I facilities are installed sufficiently far upstream on all approaches of the traffic intersection in order to send data to and receive data from any vehicle within a 200-meter range of the respective stoplines. Assuming vehicles occupy 5m of road-space on a single lane, we have an upper-limit of 40 vehicles per lane. Individual vehicle data contains speed, position and accumulated waiting time. To simplify the problem we assume constant vehicle acceleration, time-invariant travel speed, and that vehicles approaching the intersection stop before the end of queue or at the stopline. Behaviours of vehicles are limited to:

1) travelling at detected speed,

2) accelerating from standstill to the speed limit, and then travelling at the speed limit,

3) decelerating from the detected speed to standstill.

We define the following terms to facilitate further discussion:

Definition 1. The *travel time*, $x$, for a vehicle is the time it takes to travel from its current position until it leaves the stopline.

Definition 2. The *position*, $l$, of a vehicle is the distance from its current location to the stopline measured along the lane in which it is travelling.

Definition 3. The *waiting time*, $w$, of a vehicle is the accumulated time it spends at standstill.

Definition 4. The *vehicle state*, $\mathbf{z}$, is a three-dimensional vector with components speed, position and waiting time.

Definition 5. The *traffic state*, $\boldsymbol{q}$, consists of the vehicle states of all vehicles in the system

Definition 6. The *controller state*, $\mathbf{s}$, is a Boolean variable, which indicates whether the signals are green (0) or red (1)

Definition 7. The *system state*, $\mathbf{i}$, consists of the traffic state and controller state.

## 4. SYSTEM DYNAMICS

According to the Definitions in Section 3, the state of the system can be expressed as $\mathbf{i}$ { $\mathbf{q}$, $\mathbf{s}$ }, where

$$\mathbf{q}=\begin{bmatrix}\mathbf{z}(1)\\ \vdots \\ \mathbf{z}(N)\end{bmatrix}, \qquad \mathbf{s}=\begin{bmatrix}s(1)\\ \vdots \\ s(N)\end{bmatrix},$$

and $N$ denotes the total number of vehicles in the system. The vehicle state, $\mathbf{z}$, can be expressed as

$$\mathbf{z}(n)=\begin{bmatrix} v \\ l \\ w \end{bmatrix}, \text{for } n=1,2,...,N \ .$$

The Controller state, $s$, is binary, where

$$s(n)=\begin{cases} 1 & \text{if signal is green for link } n \\ 0 & \text{if signal is red for link } n \end{cases}$$

The vehicle state is related to travel time through simple dynamics

$$l=\int_0^x (v_0+at)\,dt \ , \tag{1}$$

where the initial speed, $v_0$, equals the detected speed of vehicle, i.e. $v_0 = v$. Performing the integration yields

$$l=\frac{1}{2}ax^2+v_0x \ , \tag{2}$$

Solving (2) gives

$$x=\frac{-v_0+\sqrt{v_0^2+2la}}{a} \ , \tag{3}$$

The solution obtained from (3) applies to vehicles travelling at $v_0$, accelerating or decelerating. For stopped vehicles, we sum the accumulated waiting time to (3) so that

$$x=\frac{-v_0+\sqrt{v_0^2+2la}}{a}+w \ ,$$

where $v_0 = 0$.

We present the following special cases to show how Equation (1) can be modified to obtain travel time in different situations.

**1) Free-flow to standstill**

This case applies to vehicles approaching the stopline when the corresponding signal signal has just changed to red, or vehicles approaching the end of a slow moving queue. Assuming the distance between the current position of the vehicle and its intended stopping position is $m$, then clearly $l_{min} < m <= l$, where $l_{min}$ denotes the minimum braking distance, we have

$$m=\int_0^{x'} (v_0+dt)\,dt, \text{ and } l-m=\int_0^{v^*/a} at\,dt+v^*x'' \ .$$

where $v^*$ denotes speed limit. The travel time is given by

$$x=x'+x''+\frac{v^*}{a} \ .$$

The travel time in other conditions involving a go-stop-go progression can similarly calculated.

**2) Standstill to free-flow**

This case applies to stationary vehicles responding to a green signal. We assume that a vehicle accelerates to the speed limit $v^*$ before leaving the stopline, therefore

$$l=\int_0^{v^*/a} at\,dt+v^*x' \ ,$$

$$x'=\frac{2al-v^*}{2av^*} \ ,$$

$$x=x'+w.$$

The travel time for vehicles who leave the stopline before reaching the speed limit can be inferred similarly. Having obtained the travel time for each vehicle in the system, we define a *reward function* $g(\cdot)$ as

$$g(i_t,i_{t+1})=\sum_{n=1}^{N}\left(x_n(t+1)-x_n(t)\right) , \tag{4}$$

which calculates the difference in total travel time for all vehicles in the system. We further define a *value function* $J(.)$ as

$$J(i_t)=\min_{u_t\in U_t}\left[g(i_k,i_{k+1})+\alpha J(i_{t+1})\right],\ i\in I \ , \tag{5}$$

where $\alpha$ is a discount factor. The dynamic programming formalism is then to minimize

$$\min\sum_{t=0}^{\infty}\alpha^t g_t(i_t,i_{t+1}) \ . \tag{6}$$

A DP problem as shown in Equation (6) is of the infinite horizon type. There are two iterative methods that may be used to solve Equation (6). One is *value iteration* and the other is *policy iteration*. To facilitate further discussion, we define an operator $T$, $\forall\ i \in I$, such that

$$(TJ)(i)=\min_{u_t\in U_t}\left\{g(i_t,i_{t+1})+\alpha J(i_{t+1})\right\} , \tag{7}$$

where $TJ$ produces a vector, and $TJ\,(i)$ refers to element $i$ of this vector. Let $S$ be the space that contains all value functions, then, the operator $T$ is a mapping

$$T: S\rightarrow S \ .$$

An iteration algorithm that maps $J$ to $TJ$ for $m$ iterations can be then denoted as

$$(T^mJ)(i)=\left(T(T^{m-1}J)\right)(i), \forall i\in I \ . \tag{8}$$

where for convenience we write

$$(T^0J)(i)=J_0(i),\ \forall i\in I \ .$$

For a specific policy $\mu$, we define an operator $T_\mu$ such that

$$T_\mu J=P_\mu(g+\alpha J) ,$$

for a vector $J \in S$.

Value iteration is a process that generates a sequence of $T^mJ$ starting from an initial vector $J_0$. Value iteration requires an infinite number of steps to reach the exact solution vector $J$, so a termination criterion is required

$$\left\|T^mJ-T^{m-1}J\right\|<\xi(1-\alpha)/2\alpha \ , \tag{9}$$

where $\xi$ is a specified error tolerance, $\alpha$ is a discount factor, and $||TJ||$ is the max-norm defined by:

$$\|TJ\|=\max_{i\in I}\left|(TJ)(i)\right| ,$$

The value iteration process is terminated if the condition given in Equation (9) is satisfied. The proof of convergence of value iteration can be found in [12].

Policy iteration is an alternative to value iteration and always terminates after a finite number of steps. This iteration algorithm is popular in cases where the value of a specific policy is to be found. Policy iteration starts from an initial policy $\mu_0$ and generates a sequence of new policies $\mu_1$, $\mu_2$, …, $\mu_m$. A single policy iteration step is composed of two parts. The first is the *policy evaluation* step that calculates

$$J_{\mu_m} = T_{\mu_m} J , \tag{10}$$

and the second is the *policy improvement* step in which $\mu$ is updated

$$\mu_{m+1} = \arg TJ_{\mu_m} . \tag{11}$$

This process is repeated with $\mu_{m+1}$ used in place of $\mu_m$ until we have

$$J_{\mu_{m+1}}(i) = J_{\mu_m}(i), \forall i$$

The stopping criterion shown in Equation(9) may also be used in order to reduce the number of iterations required and so speed the so;ution process. The proof that policy iteration always converges can be found in [13] and [14].

## 5. DIMESIONALITY

Dynamic programming and its iterative solutions presented in the previous section are elegant in theory, but difficult to implement. Computing (8) becomes intractable if the state-space is sufficiently large. For example, an intersection with 4 approaches, where each approach accommodates up to 40 vehicles, has a traffic state-space with dimensionality $40^4$. Assuming a vehicle may travel at any (integral) speed between 0 and 49 km, be at any (integral) location from 0 to 199 m away from stop line, wait up to 79s (in one second time steps) at a red signal, and be presented with either red or green signal; the state space has a total dimensionality of $40^4 \times 50 \times 200 \times 80 \times 2 = 4.096 \times 10^{12}$. If each state requires 4 bytes then the total storage requirement is 3814 GB. Clearly updating $J$(i) at every iteration is computationally infeasible even offline.

The dimensionality of the DP problem can reduced by using approximation techniques. We may define a continuous approximation function $\tilde{J}(\cdot,\mathbf{r}) : I \times \mathbb{R}^K \to \mathbb{R}$ to replace the exact $J(\cdot) : I \to \mathbb{R}$, where $\mathbf{r}$ is a $K$-dimensional parameter vector of $\tilde{J}$, and $I$ is the state space. By indexing successive states with positive integers, we can view the state space as the set $I = \{1, \ldots, n\}$, where $n$ is possibly infinite. The sequence of states visited by the stochastic process is denoted by $\{i_t \mid t = 0, 1, \ldots\}$. At each discrete temporal interval $t$, we have a new observation of state transition and calculate

$$u_t^*(i_t) = \arg\min_{u_t \in U} E\left\{ g(i_t, i_{t+1}) + \alpha \tilde{J}(i_{t+1}, \mathbf{r}_t) \right\} .$$

After a state transition is observed, a common technique used to update the approximation function is to find

$$\mathbf{r}^* = \arg\min_{r \in \mathbb{R}^K} \left\| J - \tilde{J} \right\| ,$$

by incrementally calculating the correction signal $\Delta r$ and updating the estimate through

$$\mathbf{r}_{t+1} = \mathbf{r}_t + \eta_t \Delta \mathbf{r}_t ,$$

where $\eta_t$ scales the adjustment to correct existing parameters. The correction signal $\Delta\mathbf{r}_t$ is usually obtained using a machine learning process. Studies on using approximate dynamic programming for adaptive traffic signal control can be found in [10] and [11]. These studies, however, use queue length as the state variable rather than vehicle speed and location. Studies on the approximate solution of the problem presented in this paper will be published shortly.

## 6. CONCLUSION

Vehicle-to-infrastructure (V2I) communication opens the opportunities to obtain precise information of vehicle speed and position. Pilot projects from car manufacturers have shown potentials for V2I techniques to improve transport efficiency and safety. Exploiting data from V2I requires new development in traffic control algorithm. In this technical note, we examined the possibility of formulating the problem in dynamic programming. We use travel time as the rewards for improving control policy, and established functions to related reward to vehicle speed, position and waiting time. We found that the dynamic programming problem thus formulated cannot be solved numerically because of the dimensionality. Approximation seems the only plausible approach to establish dynamic control using V2I communication. Numerical results from preliminary studies will be presented in future publication.

This note only explores one-way communication from vehicle to infrastructure. The V2I technologies provide two-way communication, and taking account of driver's response to controller information or instruction in control method is left for future studies.

## 7. ACKNOWLEDGMENT

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

## 8. REFERENCES

[1] Audi Travolution, http://www.audi.com.au/au/brand/en/tools/news/pool/2010/06/audi_travolution_.html

[2] BMW GREEN Wave project, http://www.themotorreport.com.au/46137/bmw-pushing-for-smarter-adaptive-traffic-lights

[3] Tarng, G.H., Chuang, B.W. 2007. Investigation of vehicle-to-infrastructure communications based on IPv6-based automotive telematics, in the *proceedings of the 7th International Conference on ITS*, *Telecommunication,* pp.1-5.

[4] Plainchault, P., Ehrlich, J., Bosch, T., Foret, S. 2003. Use of 13.56 MHz transponders for vehicle infrastructure communication, in the *proceedings of Intelligent Transport Systems*, *2003*, vol.**2**, pp.1536 – 1539.

[5] Korkmaz, G., Ekici, E., Ozguner, F. 2010. Supporting real-time traffic in multihop vehicle-to-infrastructure networks, *Transportation Research Part C*, **18**(3), pp.376 – 392.

[6] Sukuvaara, T., Nurmi, P. 2009. Wireless traffic service platform for combined vehicle-to-vehicle and vehicle-to-infrastructure communications, *Wireless Communications, IEEE*, **16**(6), pp. 54 – 61.

[7] Belanovic, P., Valerio, D., Paier, A., Zemen, T., Ricciato, F., Mecklenbrauker, C.F. 2010. On wireless links for vehicle-to-infrastructure communications, *IEEE Transactions Vehicular Technology,* **59**(1), pp. 269 – 282.

[8] Misener, J.A., Shladover, S.E. 2006. PATH investigations in vehicle-roadside cooperation and safety: a foundation for safety and vehicle-infrastructure integration research, in the *proceedings of IEEE Intelligent Transportation Systems Conference*, *2006*, Toronto, pp.9 – 16.

[9] Bellman, R., 1957. Dynamic programming, Princeton: Princeton University Press.

[10] Cai, C., Wong, C.K., Heydecker, B.G. 2009. Adaptive traffic signal control using approximate dynamic programming, *Transportation Research Part C*: *Special Issue on AI in Transport Analysis*,**17**(5), 456-474.

[11] Cai, C., Le, T. 2010. Approximate dynamic programming controller for multiple intersections, in the *proceedings of the 12th World Conference on Transport Research* (in press).

[12] Powell, W.B., 2007. Approximate Dynamic Programming: Solving the Curses of Dimensionality. John Wiley & Sons, Inc., Hoboken, New Jersey, ISBN 978-0-470-17155-4.

[13] Puterman, M.L. 1994. Markov decision process: discrete stochastic dynamic programming, John Wiley & Sons, Inc., New York, ISBN 0-471-61977-9.

[14] Bertsekas, D.P., Tsitsiklis, J.N., 1995. Neuro-Dynamic programming, Belmont, MA: Athenas Scientific.