Stochastic adaptive control model for traffic signal systems

X.-H. Yu [a] W.W. Recker [b]

[a] *Department of Electrical Engineering, California Polytechnic State University, San Luis Obispo, CA 93407, USA*
[b] *Department of Civil and Environmental Engineering, Institute of Transportation Studies, University of California, Irvine, CA 92697, USA*

Abstract

An adaptive control model of a network of signalized intersections is proposed based on a discrete-time, stationary, Markov decision process. The model incorporates probabilistic forecasts of individual vehicle actuations at downstream inductance loop detectors that are derived from a macroscopic link transfer function. The model is tested both on a typical isolated traffic intersection and a simple network comprised of five four-legged signalized intersections, and compared to full-actuated control. Analyses of simulation results using this approach show significant improvement over traditional full-actuated control, especially for the case of high volume, but not saturated, traffic demand.

Keywords: Adaptive control; Traffic signal systems; Markov processes; Traffic

1. Introduction

At a signalized intersection, traffic signals typically operate in one of three different control modes: pre-timed control, semi-actuated control and full-actuated control (Wilshire et al., 1985). In pre-timed control, all of the control parameters are fixed and preset off-line. Off-line techniques (e.g., the various versions of the TRANSYT (Robertson, 1969) family of software packages) are useful in generating the parameters for fixed timing plans for conventional pre-timed urban traffic control systems based on the deterministic traffic conditions during different time periods of the day (e.g., peak hours, off-peak hours). In actuated (both semi- and full-)control, the control signal is adjusted in accordance with a "closed-loop, on-line" control strategy based on real-time traffic demand measures obtained from detectors; while the controllers themselves respond to the fluctuations of the traffic flows in the network, the base parameters do not. Alternatively, a class of control algorithms that includes SCOOT (Split, Cycle And Offset Optimization Technique) (Hunt et al., 1982; Robertson and Bretherton, 1991) and SCATS (Sydney Coordinated Adaptive Traffic System) (Lowrie, 1982) are generally considered to be "on-line" algorithms, in which the control strategy is to "match" the current traffic conditions obtained from detectors to the "best" pre-calculated off-line timing plan.

Far fewer well-tested examples exist of real-time adaptive traffic control systems that react to actual traffic conditions on-line, the most notable among these being the well-known OPAC algorithm (Gartner, 1983), and RHODES™, a real-time traffic-adaptive signal control system that uses a traffic flow arrivals algorithm (Head, 1995) based on detector information to predict future traffic volume.

In general, two issues must be addressed to achieve real-time adaptive traffic control: (1) development of a mathematical model for the control of the stochastic, highly nonlinear traffic system, and (2) design of an appropriate control law such that the behavior of the system meets certain performance indices (e.g., minimum queue length, minimum delay time, etc.). Mathematical models used for the representation of traffic phenomena on signalized surface street networks can be classified into the following three generalized categories: (1) store-and-forward models (Hakimi, 1969; Singh and Tamura, 1974; D'Ans and Gazis, 1976), (2) dispersion-and-store models (Cremer and Schoof, 1989; Chang et al., 1994), and (3) kinematic wave models (Stephanedes and Chang, 1993; Lo, 2001).

There are two fundamental approaches for on-line optimization: binary choice logic and the sequential approach. In the binary choice logic approach, time is divided into successive small intervals, and a binary decision is made either to extend the current signal phase by one interval, or to terminate it. Examples of this approach include Miller's algorithm, traffic optimization logic (TOL), modernized optimized vehicle actuation strategy (OVA), stepwise adjustment of signal timing (SAST), etc. (Lin, 1989; Lin and Vijayakumar, 1989). The drawback of this approach is that it only considers a very short future time interval (usually 3–6 s) for the decision, and thus cannot guarantee the overall optimization of the signal operation. In the sequential approach, the length of a decision-making stage is relatively longer (from 50 to 100 s) to more closely approach the long-term optimal control. In OPAC, developed explicitly for real-time traffic control, the alternative disadvantages of the binary and sequential approaches are mitigated by incorporating a rolling-horizon approach; however, its application formally is limited to isolated intersections. Artificial neural networks (ANN) also have been applied to finding the solution for traffic control problems (Nakatsuji and Kaku, 1991) through an assumed mapping between the control variables (e.g., the split) and the objective function (e.g., the queue length); the neural network is trained off-line, using the nonlinear mapping ability of ANN, to realize this relationship. Then the signal optimization is performed on-line, using the self-organization property of an ANN. The training algorithm is a stepwise method (combination of a Cauchy machine and the "back-propagation" algorithm). However, this approach is valid only when the traffic system is in steady state.

Although most existing adaptive signal control strategies incorporate an implicit recognition that traffic conditions are time variant due to random processes, they generally adopt explicitly deterministic control models. Additionally, most employ heuristic control strategies without an embedded traffic flow model. Alternatively, the random nature of the traffic system lends itself more directly to a stochastic control approach. In the work reported here, a stochastic traffic signal control scheme, based on Markovian decision control, is introduced. The objective is to develop a real-time adaptive control strategy that explicitly incorporates the random nature of the traffic system in the control. A Markov control model is first developed; then the signal control problem is formulated as a decision-making problem for the Markov model. This approach is tested both on a typical isolated traffic intersection and a simple network comprised of five four-legged signalized intersections, and compared to full-actuated control. Analyses of simulation results using this approach show significant improvement over traditional full-actuated control, especially for the case of high volume traffic demand.

2. Markov control model

A stochastic process $x(t)$ is called Markov (Papoulis, 1984) if its future probabilities are determined by its most recent values; i.e., if for every n and $t_1 < t_2 \cdots \leqslant t_n$

$$P(x(t_n) \leqslant x_n | x(t) \forall t \leqslant t_{n-1}) = P(x(t_n) \leqslant x_n | x(t_{n-1})).$$

The adaptive control algorithm proposed is based on a discrete-time, stationary, Markov control model (also known as a Markov decision process or Markov dynamic programming) defined on $(\mathbf{X}, \mathbf{A}, \mathbf{P}, \mathbf{R})$, where

1. **X**, a Borel[2] space, is the state space and every element in the space $x \in \mathbf{X}$ is called a state;
2. **A**, also a Borel space, is defined as the set of all possible controls (or alternatives). Each state $x \in \mathbf{X}$ is associated with a non-empty measurable subset $\mathbf{A}(x)$ of **A** whose elements are the admissible alternatives when the system is in state x;
3. **P**, a probability measure space in which an element p_{ij}^k denotes the transition probability from state i to state j under alternative action k; and
4. **R**, a measurable function, also called a one-step reward.

Selection of a particular alternative results in an immediate reward and a transition probability to the next state. The total expected discounted reward over an infinite period of time is defined as

$$V \underline{\underline{\Delta}}\, E\left[\sum_{t=0}^{\infty\leftarrow} \beta^t r(x_t, a_t)\right]\leftarrow, \qquad (1)\leftarrow$$

where $r(\cdot)$ is the one-step transition reward, β^t $(0 \leqslant \beta^t \leqslant 1)$ is the discount factor, and a is the policy. The optimal reward v^*, or the supremum (least upper bound) of V, is defined as

$$v^*(x, a^*) = \sup_{a \in \mathbf{A}}[V(x, a)]. \qquad (2)\leftarrow$$

It can be obtained by solving a functional equation (also called the dynamic programming equation, or DPE):

$$v^* = Tv^*, \qquad (3)\leftarrow$$

where T is a contraction operator[3] and

$$Tv(x) = \max_{a \in \mathbf{A}}\left[q(x, a) + \beta \sum_{j=1}^{N} v(x) p_{ij}^a\right]. \qquad (4)$$

The expected one-step transition reward $q(x, a)$, is defined as

$$q(x, a) = \sum_{j=1}^{N} r_{ij}^a p_{ij}^a. \qquad (5)$$

The unique solution of the above DPE can be calculated iteratively by the successive approximation method (Hernandez-Lerma, 1989)[4]:

$$v_n(x) = \max_{a \in A}\left[q(x, a) + \beta \sum_{j=1}^{N} v_{n-1}(x) p_{ij}^a\right]. \qquad (6)$$

Therefore, for a specific control problem, once the transition matrix and the reward matrix are defined, then by maximizing the total expected reward, a policy for choosing an alternative for each state can be obtained. This represents the optimal strategy that should be followed.

3. Traffic dynamics

Consider the typical four-legged isolated traffic intersection shown in Fig. 1, where the various possible traffic movements are labeled according to NEMA (National Electrical Manufacturers Association) convention.

[2] The states of a Markov control model are defined on the Borel space—a Borel subset of a complete separable metric space. The Borel set of a metric space is the set in the smallest Borel field containing the open subset of that metric space. See, e.g., Loeve, M. Probability Theory I, fourth ed., Springer-Verlag, 1977, pp. 92.

[3] A function T from S into itself, where (S, d) is a metric space, is a contraction operator if $d(Tu, Tv) \leqslant \beta d(u, v)$ for $0 \leqslant \beta < 1$ and $\forall u \in S$, $v \in S$.

[4] See Appendix A for proof.

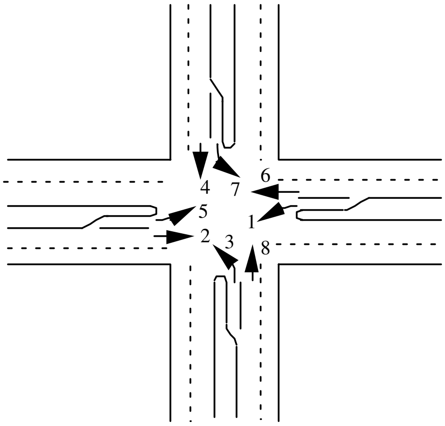

Fig. 1. A typical traffic intersection.

The state equation for the continuous traffic flow process associated with any movement j that is sampled every Δt seconds, where time is indexed with the integer k, can be expressed by the current queue $q^j(k)$:

$$q^j(k) = q^j(k-1) + \Delta q^j(k), \quad j = 1, 2, \ldots, 8, \qquad (7)\leftarrow$$

where $\Delta q^j(k) = q^j_{\text{in}}(k) - q^j_{\text{out}}(k)$ is the difference between the input $q^j_{\text{in}}(k)$ and the output $q^j_{\text{out}}(k)$ during time interval $[k-1, k)$, and $q^j(k-1)$ is the queue at previous time instant $(k-1)$. For a typical four-legged traffic intersection with eight movements, the current queue $q(k)$ can be further defined by the vector

$$\underline{q}(k) = [q^j(k)]' = [q^1(k), q^2(k), \ldots, q^8(k)]'^{\leftarrow}, \qquad (8)\leftarrow$$

where prime ($'$) is used to denote transpose. The input $\underline{q_{\text{in}}}(k)$ and output $\underline{q_{\text{out}}}(k)$ of the intersection (i.e., number of vehicles entering/leaving the intersection) can also be similarly defined as vectors of like dimension:

$$\underline{q_{\text{in}}}(k) = [q^j_{\text{in}}(k)]'^{\leftarrow}, \quad \underline{q_{\text{out}}}(k) = [q^j_{\text{out}}(k)]'^{\leftarrow}. \qquad (9)$$

The output $\underline{q_{\text{out}}}(k)$ can further be expressed as a function of the current control of the intersection, $u(k)$, and the current queue, $\underline{q}(k)$:

$$\underline{q_{\text{out}}}(k) = \underline{f_{\text{out}}}(u(k), \underline{q}(k)), \qquad (10)\leftarrow$$

where $\underline{f_{\text{out}}}(k)$ is also a vector of the same dimension, i.e.,

$$\underline{f_{\text{out}}}(k) = [f^j_{\text{out}}(k)]'^{\leftarrow} \qquad (11)\leftarrow$$

and where the elements $f^j_{\text{out}}(k)$ are determined by

$$f^j_{\text{out}}(k) = \begin{cases} \min\left[q^j(k); \frac{\Delta t}{h_{\min}}\right], & u^j(k) = 0, \\ 0, & u^j(k) = 1 \end{cases} \qquad (12)\leftarrow$$

in which $h_{\min}$ is the minimum headway, and $u^j(k)$ is a dichotomous variable indicating the control signal for the jth movement: $u^j(k) = 0$ denotes that the jth movement has the green signal and $u^j(k) = 1$ indicates a red signal.

Under standard eight-phase dual-ring control (Fig. 2), the barrier divides the eight NEMA phases into two interlocked groups (rings): east/west and north/south; in each ring, four movements (two through movements and their corresponding left-turn movements) must be served if there is demand. Although there are $2 \cdot 4! = 48$ different phase sequences available, depending on the traffic demand, the ring and barrier rules restrict the maximum number of phase transitions in a single cycle to six—a maximum of three distinct phase combina-

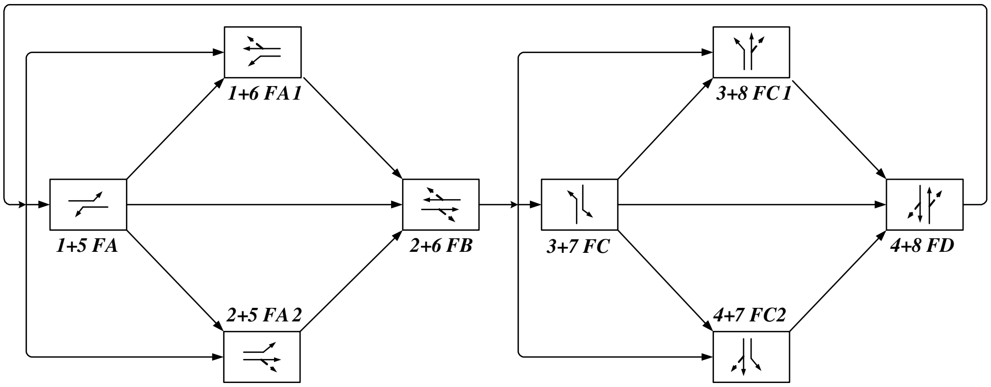

Fig. 2. Eight-phase dual-ring signal control.

tions on each side of the barrier. Using this information, the phase sequencing constraints on choice of the current control depends, at most, on three previous control signals:

$$u(k) = f_u(q(k), \tau, u(k-\tau_1), u(k-\tau_2), u(k-\tau_3)), \qquad (13)\leftarrow$$

where τ_1 is the time duration of the most recent previous phase, τ_2 is the time duration of the next-to-last phase, and so on. In addition to the sequencing constraints, the duration of the current signal, τ, must be bounded between some minimum (e.g., minimum green, minimum green extension) and maximum (e.g., maximum green) time period:

$$\tau_{\min} \leqslant \tau \leqslant \tau_{\max}. \qquad (14)\leftarrow$$

This schema easily can be generalized to traffic networks with multiple intersections. In a traffic network with n intersections, the order of the dynamic equations is increased to $n \times 8$ (assuming that there are eight traffic movements in each intersection). However, any complicated traffic network can be decomposed into a group of small "elementary networks", as shown in Fig. 3, consisting of five intersections. In this manner, the study of the entire traffic network can be reduced to the analysis of these elementary networks and the inter-connections between them.[5]

The complete traffic dynamics model for the network shown in Fig. 3 includes the following equations:

$$\begin{aligned}
&\underline{u}(k) = [\underline{u}_1(k), \underline{u}_2(k), \ldots, \underline{u}_5(k)]'^{\leftarrow}, \\
&\underline{f_u}(k) = [f_{u1}(k), f_{u2}(k), \ldots, f_{u5}(k)]'^{\leftarrow}, \\
&\underline{q_{\text{out}}}(k) = \left[\underline{q_{\text{out}}}_1(k), \underline{q_{\text{out}}}_2(k), \ldots, \underline{q_{\text{out}}}_5(k)\right]'^{\leftarrow}, \\
&\underline{q_{\text{in}}}(k) = \left[\underline{q_{\text{in}}}_1(k), \underline{q_{\text{in}}}_2(k), \ldots, \underline{q_{\text{in}}}_5(k)\right]'^{\leftarrow}, \\
&\underline{q}(k) = \left[\underline{q}_1(k), \underline{q}_2(k), \ldots, \underline{q}_5(k)\right]'^{\leftarrow}, \\
&\underline{f_{\text{out}}}(k) = \left[\underline{f_{\text{out}}}_1(k), \underline{f_{\text{out}}}_2(k), \ldots, \underline{f_{\text{out}}}_5(k)\right]'^{\leftarrow}, \\
&\underline{q}(k) = \underline{q}(k-1) + \underline{\Delta q}(k),
\end{aligned} \qquad (15)\leftarrow$$

where

$$q^j_{\text{out}i}(k) = f_{\text{out}}(u^j_i(k), q^j_i(k)), \qquad (16)\leftarrow$$

[5] This approach also facilitates parallel processing techniques to improve the computational efficiency for real-time control.

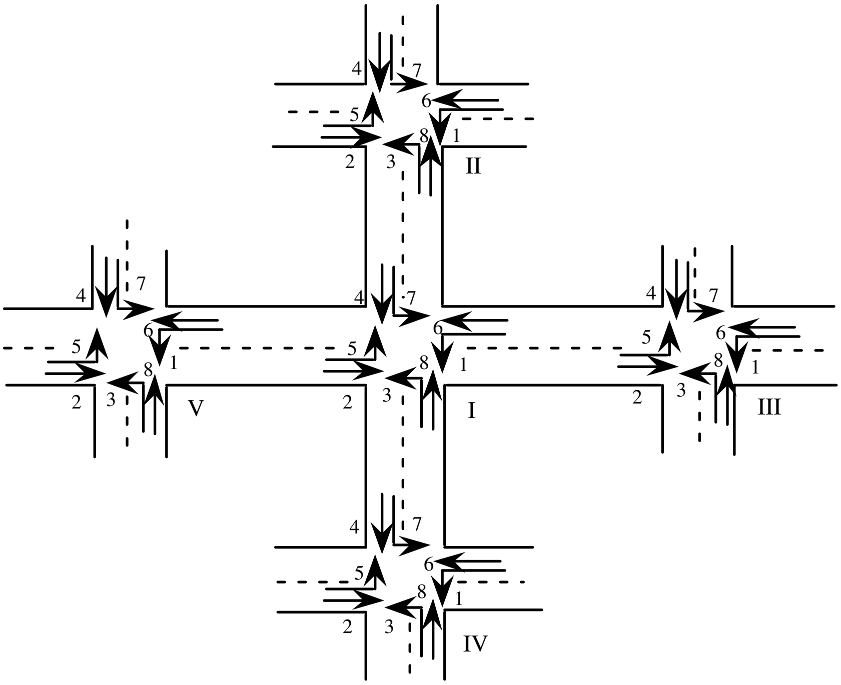

Fig. 3. A typical elementary traffic network with five intersections.

$$f_{\text{out}}^{j}(k)=\begin{cases}\min\left[q_i^j(k);\frac{\Delta t}{h_{\min}}\right], & u_i^j(k)=0\\ 0, & u_i^j(k)=1\end{cases}\qquad i=1,2,\ldots,5;\; j=1,2,\ldots,8. \qquad (17)\leftarrow$$

In Eq. (15), the subscripts to the various vector quantities refer to the particular intersection, and the vector quantities themselves are as previously defined.

Unlike the case of an isolated intersection, the interactions between intersections must be included in the traffic model for this case. For example, consider the simple case of the two adjacent intersections shown in Fig. 4.

The eight traffic movements associated with each intersection can be classified into two different types:

1. *External movement.* The arrival vehicles come from/go to a "dummy node" outside the network (these vehicles can be considered as the "input/output" of this network); and

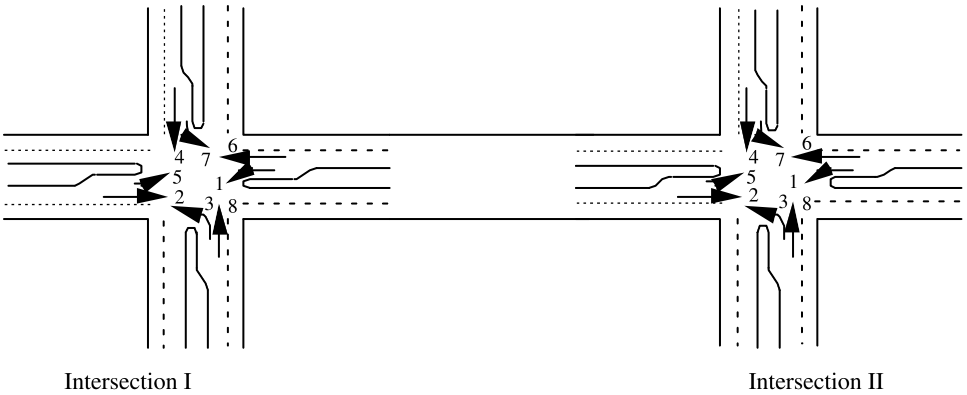

Fig. 4. A traffic network with two intersections.

2. *Internal movement.* The arrival vehicles come from/go to a neighboring node inside the network (these vehicles can be considered as the "interconnection" of this network).

For example, movements 1 and 6 are internal movements of intersection I, which receive the outputs from intersection II, movements 3 and 6. All of the other movements of intersection one are external movements. Similarly, all of the movements of intersection II are external movements, with the exception of movements 2 and 5, which receive the output from the movements 2 and 7 of intersection I. Then, for intersection I, the internal movements are defined by

$$\begin{aligned} q^1_{\mathrm{inI}}(k) &= f_{\mathrm{in}}\left(\beta^{3,1}_{\mathrm{II,I}}\left(k - T^3_{\mathrm{II,I}}\right), q^3_{\mathrm{outII}}\left(k - T^3_{\mathrm{II,I}}\right), \beta^{6,1}_{\mathrm{II,I}}\left(k - T^6_{\mathrm{II,I}}\right), q^6_{\mathrm{outII}}\left(k - T^6_{\mathrm{II,I}}\right)\right), \\ q^6_{\mathrm{inI}}(k) &= f_{\mathrm{in}}\left(\beta^{3,6}_{\mathrm{II,I}}\left(k - T^3_{\mathrm{II,I}}\right), q^3_{\mathrm{outII}}\left(k - T^3_{\mathrm{II,I}}\right), \beta^{6,6}_{\mathrm{II,I}}\left(k - T^6_{\mathrm{II,I}}\right), q^6_{\mathrm{outII}}\left(k - T^6_{\mathrm{II,I}}\right)\right), \end{aligned} \qquad (18)\leftarrow$$

where $\beta^{j_1,j_2}_{i_1,i_2}$ is defined as the vehicle turning fraction from intersection i_1, movement j_1 to intersection i_2, movement j_2,[6] and where $T^{j_1}_{i_1,i_2}$ represents the travel time for the first vehicle in the platoon of vehicles in movement j_1 of intersection i_1 to reach intersection i_2.

The time-dependent turning factors can be represented by the turning fraction matrix, $\underline{\boldsymbol{\beta}}(k)$, whose elements indicate the percentage of vehicles turning from a certain movement at the upstream intersection to a specific movement at the down stream intersection. For the case of two intersections shown in Fig. 4, $\underline{\boldsymbol{\beta}}(k)$ can be written as a 16 × 16 matrix:

$$\underline{\boldsymbol{\beta}}(k) = \left[\begin{array}{c:c} \underline{0} & \left[\beta^{j_1j_2}_{\mathrm{I,II}}(k)\right] \\ \hdashline \left[\beta^{j_1j_2}_{\mathrm{II,I}}(k)\right] & \underline{0} \end{array}\right] = \left[\begin{array}{c:c} \underline{0} & \underline{\boldsymbol{\beta}_{\mathrm{I,II}}}(k) \\ \hdashline \underline{\boldsymbol{\beta}_{\mathrm{II,I}}}(k) & 0 \end{array}\right], \qquad (19)\leftarrow$$

where

$$\underline{\boldsymbol{\beta}_{\mathrm{I,II}}}(k) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \beta^{2,2}_{\mathrm{I,II}}(k)\leftarrow & 0 & 0 & \beta^{2,5}_{\mathrm{I,II}}(k)\leftarrow & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \cdots\leftarrow & & & & \cdots\leftarrow & & & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \beta^{7,2}_{\mathrm{I,II}}(k)\leftarrow & 0 & 0 & \beta^{7,5}_{\mathrm{I,II}}(k)\leftarrow & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\underline{\boldsymbol{\beta}_{\mathrm{II,I}}}(k) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \beta^{3,1}_{\mathrm{II,I}}(k)\leftarrow & 0 & 0 & 0 & 0 & \beta^{3,6}_{\mathrm{II,I}}(k)\leftarrow & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \beta^{6,1}_{\mathrm{II,I}}(k)\leftarrow & 0 & 0 & 0 & 0 & \beta^{6,6}_{\mathrm{II,I}}(k)\leftarrow & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Using this general expression for $\boldsymbol{\beta}(k)$

$$q^j_{\mathrm{in}\,i}(k) = f_{\mathrm{in}}\left(\beta^{r,j}_{m,i}(k - T^r_{m,i}), q^r_{\mathrm{out}\,m}(k - T^r_{m,i})\ \forall r \in M^i_m, m \in I_i\right), \qquad (20)$$

[6] For example, $\beta^{3,1}_{\mathrm{II,I}}(k)$ represents the percentage of vehicles from movement 3 of intersection II turning to movement 1 of intersection I during time interval k.

where I_i is the set of all neighboring intersections with direct approaches to intersection i, and M^i_m is the set of all movements of intersection m that contribute to the internal movements of intersection i.

Practical application of Eq. (20) relies on the ability to predict both the time-dependent turning fractions, $\underline{\beta}(k)$, and the platoon travel times from neighboring intersections to the target intersection, $T^{j_1}_{i_1,i_2}$. The estimation of turning fractions from count data has been the subject of numerous investigations; see, e.g., the review provided by Maher (1984) for a summary of models that require counts for only one cycle but need prior turning proportion estimation. However, the accuracy of such methods is highly dependent on how representative the *a priori* estimates are of the current events. Alternatively, estimations based on simple time-series analyses do not need such prior estimates but require a long time frame which impedes their responsiveness, and are unreliable during times of sudden and highly irregular turning movement changes caused by such unforeseen events as traffic accidents. Davis and Lan (1995) have proposed a method that estimates intersection turning movement proportions from less-than-complete sets of traffic counts, even under conditions in which the number or placement of detectors does not support complete counting. Chang and Tao (1997) propose a time-dependent turning estimation that incorporates signal timing parameters on the distribution of intersection flows. More recently, Mirchandani et al. (2001) propose four closed-form estimation methods: (1) maximum entropy (ME), (2) generalized least-squared (GLS), (3) least-squared error (LS), and (4) least-squared error/generalized least-squared error (LS/GLS). Although not specifically addressing the estimation of turning fractions for purposes of signal timing, Chen et al. (2005) and Nie et al. (2005) have examined a generalized path flow estimator (PFE) as a one-stage network observer to estimate path flows and path travel times from traffic counts in a transportation network, and have shown it to be a reasonably accurate method for estimating dynamic path flows based on limited real-time detector data. The estimated path flows can further be aggregated to obtain dynamic origin–destination (O–D) flows, a by-product of which are the turning fractions at the various nodes in the network. In the results presented here, we presume that the $\underline{\beta}(k)$ can be determined from one or another of these existing estimation procedures.

In order to determine the platoon travel times from neighboring intersections to the target intersection, $T^{j_1}_{i_1,i_2}$, we employ the well-known empirical model developed by Robertson (1969) for platoon dispersion to describe the flow dynamics from upstream intersections to downstream movements. Robertson's dispersion model has been used and tested extensively in field applications involving both TRANSYT and SCOOT, and found to be a very effective representation of platoon dynamics. In its basic form, the model has the representation:

$$Q_1(t_0+T) = F\cdot Q_2(t_0) + (1-F)\cdot Q_1(t_0+T-1), \qquad (21)\leftarrow$$

where

$$F = \leftarrow\frac{1}{1+\alpha\beta T_{\text{avg}}}$$

and where Q_1, Q_2 are the traffic volumes at the downstream and upstream intersections (measured in vehicles/h), respectively; α and β are called platoon dispersion parameters; t_0 is the initial time when the platoon leaves the upstream intersection; T_{avg} is the average travel time, and T is the minimum travel time between the two intersections, i.e., the time for the lead vehicle in the platoon to reach the downstream intersection.[7] T is related to T_{avg} through the parameter β, i.e.,

$$T = \beta T_{\text{avg}}. \qquad (22)\leftarrow$$

Substituting Robertson's platoon dispersion formula into Eq. (20) leads to

$$q^j_{\text{in}\,i}(k) = \leftarrow\sum_{\substack{\forall r\in M^i_m\\ m\in I_i}}\left(F\cdot \beta^{r,j}_{m,i}(k-T^r_{m,i})\cdot q^r_{\text{out}\,m}(k-T^r_{m,i}) + (1-F)\cdot q^j_{\text{in}\,i}(k-1)\right. \qquad (23)\leftarrow$$

with the current control vector defined by

$$u_i(k) = f_u(q_i(k),\tau_i,u_i(k-\tau_1),u_i(k-\tau_2),u_i(k-\tau_3)), \qquad (24)\leftarrow$$

[7] Both T and T_{avg} must be rounded to integer values.

where $\tau_{\min i} \leqslant \tau_i \leqslant \tau_{\max i}$, $\beta^{r,j}_{m,i}(\cdot)$←can be derived from counts from upstream stopline detectors according to existing procedures discussed previously, and where the $T^r_{m,i}$ are determined by Eq. (22) from parameter specification and average travel speed.

4. Markov adaptive control model for traffic signal control

The state variable in the traffic dynamics equation developed above is queue length. Although the state of the Markov control model can be defined as the number of vehicles in the intersection, this approach results in an excessively large number of states, even for a single intersection.[8] To address this problem, the state of the Markov control model is instead defined by introduction of a binary threshold value (number of vehicles) indicating whether or not the current queue for a particular movement is sufficiently large to be "congested", i.e., if the queue length of a specific movement is greater than its threshold value, then the movement is in the "congested mode"; otherwise it is in the "non-congested mode". These binary modes (congestion/non-congestion) are defined as the two states in the state space $\mathbf{X}$.[9]

Since the state space is discrete, the probability measure $\mathbf{P}$ is a discrete transition law, and the probability matrix $\underline{\mathbf{P}}$ is time-varying due to the time-varying traffic flow. At time step k, $\underline{\mathbf{P}}$ is a function of $\underline{q}(k)$, $\Delta\underline{\hat{q}}(k+1)$, and $\underline{u}(k)$:

$$\underline{\mathbf{P}}(k) = \underline{f_P}[q(k), \underline{q_{\text{in}}}(k+1), u(k)], \qquad (25)←$$

where $\underline{q}(k)$ is the current queue, $\Delta\underline{\hat{q}}(k+1)$←is the estimated number of arrivals in the next time interval, and $\underline{u}(k)$ is the control signal. Assuming that at time step k, the current queue length of a specific movement i is denoted by q_0; and q_{g} vehicles can pass through the intersection if the traffic signal for this movement is green; then the transition probability from any current state (either congested or non-congested) to the non-congested state under control signal u can be written as

$$p^{u_i}_{S_i \to N_i} = p\left(\hat{q}^i_{\text{in}} + q^i_0 - \delta(u_i) \cdot q^i_{\text{g}} \leqslant q^i_{\text{threshold}}\right) \qquad (26)$$

and, to the congested state, as

$$p^{u_i}_{S_i \to C_i} = 1 - p^{u_i}_{S_i \to N_i}, \qquad (27)$$

where

$$\delta(u_i) = \begin{cases} 1, & \text{when } u_i = G_i, \\ 0, & \text{otherwise.} \end{cases} \qquad (28)←$$

In the above, $q^i_{\text{threshold}}$ is the threshold which defines the congested/non-congested state; S_i is the current state (N_i for non-congested state and C_i for congested state); u_i is the control signal (G_i for green signal and R_i for red signal). Two special cases are noted in that:

$$p^{R_i}_{C_i \to C_i} \equiv 1, \quad \text{and} \quad p^{R_i}_{C_i \to N_i} \equiv 0. \qquad (29)$$

As mentioned previously, for a typical traffic intersection with eight independent movements, the total number of states is $2^8 = 256$. The transition probability for each movement is also independent; therefore, the overall transition probability for an intersection is

$$p^{u}_{\text{State}_j \to \text{State}_r} = \prod_{i=1}^{8} p^{u_i}_{S_i \to S_i}, \qquad (30)$$

where $j, r = 1, 2, \ldots, 256$; and $\underline{u}(k) = [u_1, u_2, \ldots, u_8]'$.←

[8] For example, if the number of vehicles under consideration is 20 per movement, then for an isolated intersection with eight movements, the total number of states is $21^8 \approx 3.78 \times 10^{10}$.

[9] For an isolated intersection with eight movements and ten vehicles per movement, the number of states is dramatically reduced by a factor of 10^8 to $2^8 = 256$.

The reward matrix $\underline{\mathbf{R}}$ has the same dimension and a definition similar to that of the probability matrix. The control objective is to maintain the non-congested condition or, if already congested, to transit to a non-congested state. The latter yields a greater reward than the former and the transition from a non-congested state to congested carries a greater penalty than remaining in a congested state. Since the congested/non-congested state is defined in terms of queue length, the reward matrix is a function of the current queue, the threshold, and the control signal:

$$\underline{\mathbf{R}}(k) = \underline{f_r}[\underline{q}_0(k), \underline{q}_{\text{threshold}}(k), \underline{u}(k)]. \qquad (31)\leftarrow$$

For example, if the objective is to minimize the queue length, then the reward for each possible case can be chosen as the following:

$$\begin{aligned}
r^{G_i}_{N_i\to N_i} &=\leftarrow q^i_0 + M_1,\\
r^{R_i}_{N_i\to N_i} &=\leftarrow q^i_0 + M_2,\\
r^{G_i}_{N_i\to C_i} &=\leftarrow q^i_0 + M_3,\\
r^{R_i}_{N_i\to C_i} &=\leftarrow M_4,\\
r^{G_i}_{C_i\to N_i} &=\leftarrow q^i_0 + M_5,\\
r^{R_i}_{C_i\to N_i} &=\leftarrow N.A.,\\
r^{G_i}_{C_i\to C_i} &=\leftarrow q^i_0 + M_6,\\
r^{R_i}_{N_i\to N_i} &=\leftarrow M_7,
\end{aligned}$$

where M_i, $i = 1, 2, \ldots, 7$, are constants which can be specified for a specific traffic control problem.[10]

Similar to the probability matrix, the overall reward for an intersection with eight independent movements is

$$r^{u}_{\text{State}_j\to\text{State}_r} =\leftarrow \prod_{i=1}^{8} \Bigg(r^{u_i}_{S_i\to S_i}, \qquad (32)$$

where $j, r = 1, 2, \ldots, 256$.

The signal phases are the different alternatives for each state; for a typical isolated traffic intersection with eight independent movements under eight-phase dual-ring signal, the signal control problem takes the form of a 256-state Markov process with eight alternatives for each state. The optimal policy is then obtained by selecting the alternative for each state that maximizes the total expected reward. As has been demonstrated above, this optimal solution is unique and can be calculated iteratively by the successive approximation method.

The proposed Markov control model can be illustrated by the simplified example of the two-phase isolated intersection shown in Fig. 5, in which traffic flows along two directions, i.e., north/south (denoted by 1) and east/west (denoted by 2). Thus, there are four possible states, i.e., N_1N_2, N_1C_2, C_1N_2, and C_1C_2. Fig. 6 shows the schematics of this Markov chain. To simplify the example, amber displays and all red signals (R_1R_2) are ignored; G_1G_2 is prohibited for obvious reasons. Under these conditions, there are two alternatives (signal phases) in each state, i.e., G_1R_2 and R_1G_2. With the usual assumption of Poisson arrivals, the various transition probabilities can be calculated directly. For example, the transition probabilities from the non-congested state are

$$p^{G}_{N\to N} =\leftarrow \sum_{n=1}^{q_{\text{threshold}}-q+q_g} \frac{(\lambda\Delta t)^n \mathrm{e}^{-\lambda\Delta t}}{n!}, \quad p^{R}_{N\to N} =\leftarrow \sum_{n=1}^{q_{\text{threshold}}-q} \Bigg(\frac{(\lambda\Delta t)^n \mathrm{e}^{-\lambda\Delta t}}{n!}, \qquad (33a)$$

$$p^{G}_{N\to C} =\leftarrow 1 - p^{G}_{N\to N}, \quad p^{R}_{N\to C} =\leftarrow 1 - p^{R}_{N\to N}, \qquad (33b)\leftarrow$$

where n is a positive integer ($n = 1, 2, \ldots$); λ is the average vehicle arrival rate (vehicles/h) and Δt is the time interval (i.e., duration of each counting period).

[10] If the objective is to minimize the delay time, the specific rewards can also be chosen as functions of the vehicle delay.

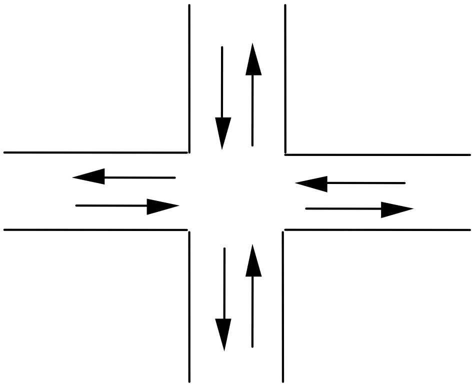

Fig. 5. An isolated intersection with through movements only.

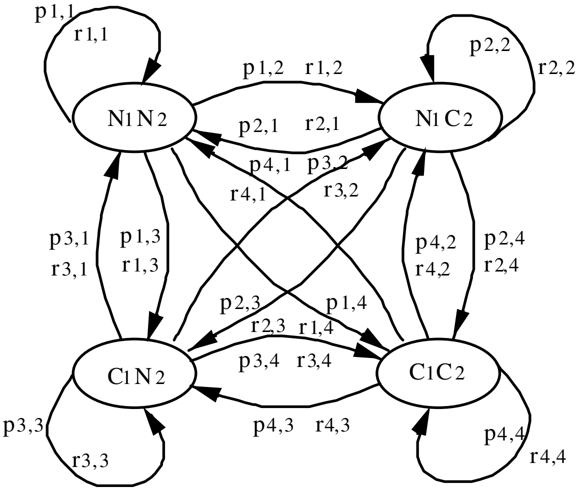

Fig. 6. The Markov chain for the example.

The corresponding state probabilities are

$$\begin{aligned}
p^{G_1R_2}_{N_1N_2\to N_1N_2} &= p^{G_1}_{N_1\to N_1}\cdot p^{R_2}_{N_2\to N_2},\\
p^{G_1R_2}_{N_1N_2\to N_1C_2} &= p^{G_1}_{N_1\to N_1}\cdot p^{R_2}_{N_2\to C_2},\\
&\vdots
\end{aligned}\qquad (34)\leftarrow$$

Since, for this example, there are four states with two alternatives for each state, the elements above form an 8×4 transition probability matrix, as shown in Table 1. Elements of the reward matrix can be calculated in similar fashion.

A general block diagram of traffic control using this scheme at an isolated signalized intersection is illustrated in Fig. 7, and a corresponding computational flow chart shown in Fig. 8. Based on the current and the estimated traffic flow, the controller generates a traffic control signal to control the traffic system for the next time interval.

In the application of this procedure to real-time adaptive control for a traffic system, the time-varying probability matrix $\underline{\mathbf{P}}$ and the reward matrix $\underline{\mathbf{R}}$ are calculated and updated every Δt seconds[11]; a decision is then

[11] The minimum time interval is chosen as $\Delta t = \tau_{\min i}$ (i.e., minimum green extension time).

Table 1
The state probability matrix for the example

State		N_1N_2	N_1C_2	C_1N_2	C_1C_2
N_1N_2	G_1R_2	$p^{G_1R_2}_{N_1N_2\to N_1N_2}$	$p^{G_1R_2}_{N_1N_2\to N_1C_2}$	$\cdots\leftarrow$	$p^{G_1R_2}_{N_1N_2\to C_1C_2}$
	R_1G_2	$p^{R_1G_2}_{N_1N_2\to N_1N_2}$			
N_1C_2	G_1R_2		$\ddots$		
	R_1G_2		$\ddots$		
C_1N_2	G_1R_2			$\ddots$	
	R_1G_2			$\ddots$	
C_1C_2	G_1R_2				$p^{G_1R_2}_{C_1C_2\to C_1C_2}$
	R_1G_2				$p^{R_1G_2}_{C_1C_2\to C_1C_2}$

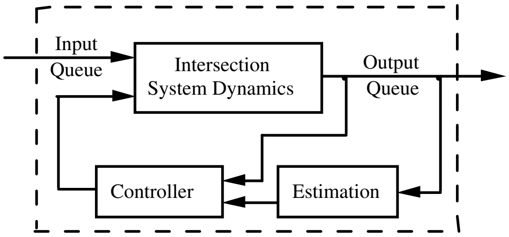

Fig. 7. Traffic control at signalized intersection.

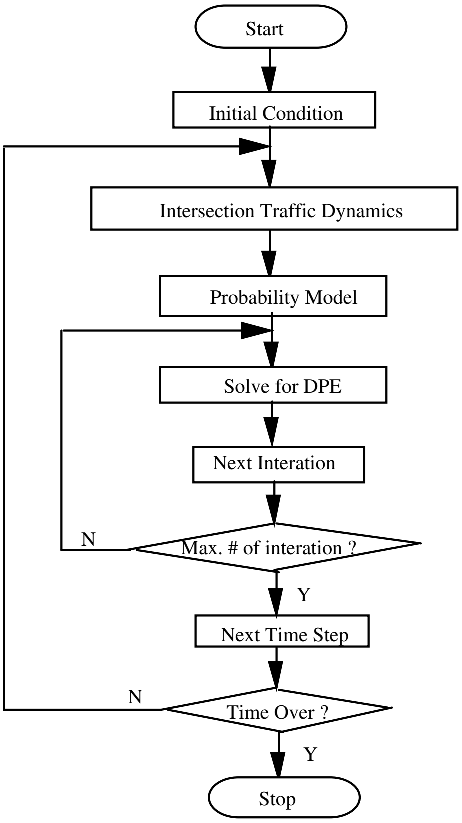

Fig. 8. Computational flow chart.

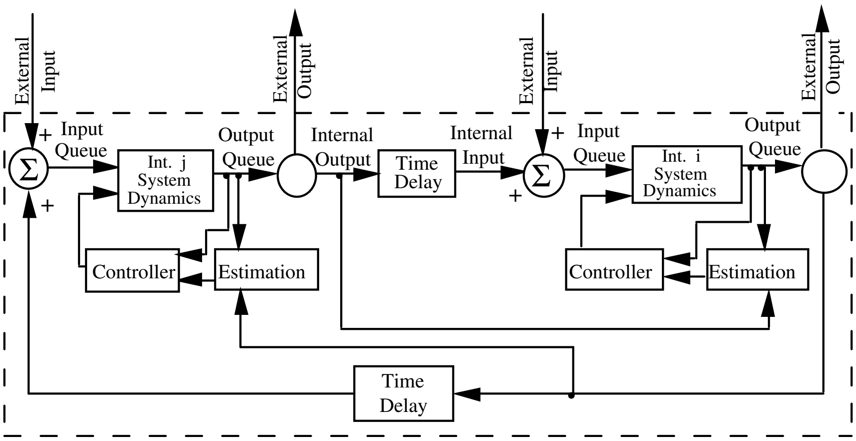

Fig. 9. Traffic control for two intersections.

made regarding the choice of the control signal for the next time interval based on the current measurement from the detector, as well as the estimation. Once the optimal policy is found, it is implemented for one time step (i.e., Δt seconds). At the next time interval, both the probability matrix and reward matrix are updated and the whole decision-making process is repeated.

To enforce the phase constraints, a step-by-step decision-making procedure (also termed a "decision tree") is employed. For example, a decision is made first to determine which ring will be served by the Markovian decision algorithm. After this is determined, the second decision is to choose one of the four alternatives from the first decision, again using the Markovian decision algorithm. The next phase is either fixed or can be chosen from the two phases left, depending upon the second decision. At the last decision step for this ring, there is either no phase or just one fixed phase left. This procedure not only guarantees the phase constraints but also dramatically reduces computation time.

Application of the decision control to the signal control of a network of multiple intersections proceeds along a similar manner; a block diagram for the control system of two traffic intersections is shown in Fig. 9. In such cases, the control signal of the two neighboring intersections do not interact until some minimum travel time, at which time the control is modeled through the probability estimation of internal movement arrivals at the downstream intersections. That is, assuming that the minimum travel time between two intersections is longer than the minimum green extension time, the control signals of the two intersections do not interact due to the random travel time delay between them. After the minimum travel time, the control at one intersection does affect intersections downstream; this effect is modeled in the probability estimation at the downstream intersections. As a result, adjacent intersections can be "isolated" and the respective control actions can be calculated separately.

5. Results on application of Markov adaptive signal control model

In this section, the control model is tested by simulation on both an isolated traffic intersection and a typical traffic network with five interconnected intersections to evaluate its performance with respect to conventional full-actuated control. Specifically, a series of computer simulations are performed, under various different vehicle arrival rates, and the means and variances of the respective performance measures of the conventional and proposed adaptive control algorithm are analyzed. The simulations assume that queues on all approaches are empty as an initial condition and that vehicle arrivals on external approaches follow a Poisson distribution; for demonstration purposes, a value of $q_{\text{threshold}} = 1$ (i.e., the presence of any queue) was assumed. The reward matrix was based on the objective being to minimize the queue length, and the reward calculated according to Eq. (32). In the case of the network simulation, the distance between any two adjacent intersections is chosen to be 1000 feet. The parameters used in the simulation (for all the movements) are summarized as follows:

Parameter	Value
Minimum green time (s)	3
Maximum green time (s)	30
Extension (gap) time (s)	3
Yellow time (s)	3
All red time (s)	0
Loss time (s)	0
Minimum departure headway (s)	2
Minimum arrival headway (s)	2

Using the same set of input (arrival) data, the Markovian control algorithm and the conventional full-actuated control were applied to a four-legged isolated traffic intersection, such as that shown in Fig. 1, with eight movements (four through movements and their corresponding left-turn movements) to evaluate their performances. The algorithm used to simulate full-actuated control was designed to mimic the logic of a common Type 170 dual ring controller with parameters as specified in the previous table—eight-phase operation was assumed. To minimize initial condition effects, the two algorithms are applied for a simulated time of 65 min, and the average delay (per vehicle) during the last five minutes of the simulation is used for comparison.[12] Two different general cases were considered: (1) uniform (balanced) demand among all conflicting movements, and (2) the through traffic demand dominates the left-turn demand by a ratio of 2:1. The two algorithms were applied for different arrival rates, representing a range of both unsaturated and saturated conditions. (Under the assumption of 2-second minimum headways, the intersection has a total capacity of 3600 vehicles per hour of green.) In order to provide statistical significance for the simulation results, the two algorithms were tested on different sets of random data for each arrival rate (a total of forty in the cases in which left-turning traffic was assumed equal to through traffic, and fifteen in the cases in which left-turning traffic was equal to half of the through traffic).

The means of the average delay per vehicle for the final five-minute period of each set of forty simulations corresponding to the two cases of left-turn to through traffic ratios of 1.0 (LT/T = 1.0) and 0.5 (LT/T = 0.5) are plotted in Fig. 10, where "MAC" stands for the Markov adaptive control algorithm, and "FAC" stands for the full-actuated control. As a further "benchmark" comparison, delay calculations based on Webster's delay equation for Poisson arrivals under fixed-time (pre-timed) control are also provided (labeled Pre ($C = 60$ s) and Pre ($C = 45$ s) for cycle lengths of 60 and 45 s, respectively).

Significance tests based on *t*-statistics resulting from hypothesis tests on the difference of sample means indicate that the difference in means of the simulation results is significantly different (at 0.05 level or above) for all cases except for the LT/T = 1.0 case in which the total intersection volume is 1500 vph. The hypothesis tests on the difference of means assume that the two populations are independent and have a normal distribution. Alternatively, order statistics (distribution-free statistics) estimate the limits within which a certain percentage of the probability of the random variable lies with a certain degree of confidence without having prior knowledge of the probability distribution. For the case involving 40 samples taken from a population, the upper/lower bound within which 90% of the probability of the random variable lies can be obtained with 92% confidence. Fig. 11 displays these bounds on the steady state delay resulting both from full-actuated and from Markovian control algorithms.

From the above figures, except for the case in which the left-turn traffic volume is equal to the through volume (LT/T = 1.0) and the traffic volume is relatively light (e.g., arrival rate is 200 vehicles/hour/movement), the performance of the Markov algorithm is significantly better than the fully actuated controller (as well as the pre-timed controller). For example, for LT/T = 1.0, when $\lambda = 300$, the Markov algorithm shows about a 25% improvement on the average steady state delay; for $\lambda = 400$ and $\lambda = 500$, the average steady state delay of the Markov controller is only about one half of that of the full-actuated controller. As expected, under saturated conditions both algorithms exhibit increasingly worse delays, although the Markov control (on aver-

[12] In most cases, the steady state is reached within an hour of simulation.

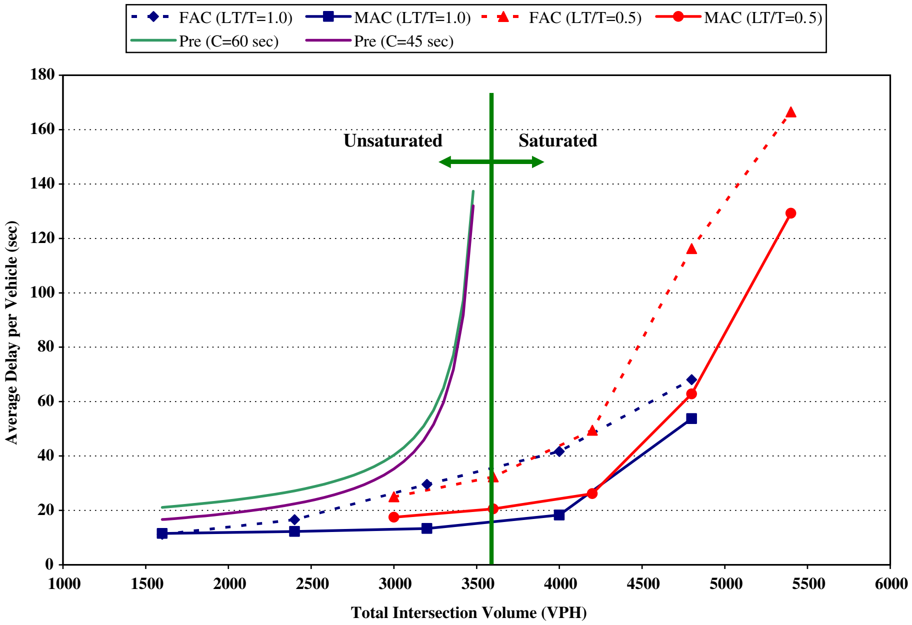

Fig. 10. Algorithm performance comparison for isolated intersection.

age) still outperforms full-actuated control. The simulation results indicate that by applying the Markov adaptive control algorithm, the average delay at an isolated intersection may be reduced dramatically (22–51%).

The Markov adaptive control algorithm was also tested on a typical traffic network of five intersections, such as that depicted in Fig. 3. For this case, Poisson arrivals were assumed at the external inputs; the arrivals at all internal approaches are an outcome of the control strategy employed at associated upstream intersections. The tests were conducted for LT/T = 1.0 using five different arrival rates: $\lambda = 200$, 300, 400, 500 and 600 vehicles per hour per movement. The internal approaches linking the five intersections were assumed to be 1000 ft in length, and the average travel speed assumed to be 30 mph (resulting in a value of $T_{\text{avg}} = 23$ s). The parameters in Robertson's platoon dispersion model were assumed to be $\alpha = 0.35$, $\beta = 0.8$—the common default values for US studies. The mean values (of the 40 sets of data) of the steady state delay are plotted in Fig. 12. The dotted lines in Fig. 12 display the upper/lower bounds within which 90% of the probability of the steady state delay resulting both from full-actuated and from Markovian control algorithms lay.

The results indicate that the Markov algorithm substantially outperforms traditional full-actuated control, particularly when the intersection is at, or near saturation. For example, when $\lambda \leqslant 500$ (total intersection volume of 4000 vph), the average steady state delay of the Markov controller is only about one half of that of the fully actuated controller. Under heavy over-saturated conditions ($\lambda = 600$), delay with both algorithms tend to converge at a relatively high value.

We note that, under simple five-node network conditions with identical arrival rates, the performance of the Markov control algorithm closely mirrors that obtained in the case of the isolated intersection example (Fig. 13). Although preliminary, the results suggest that application of the algorithm in a network setting tends to decrease variability in performance; this is expected, since the variability expressed in the Poisson arrival patterns at the external nodes becomes an increasingly minor factor as the number of internal approaches increases. This latter factor may help to explain the large variance seen in the isolated intersection case under heavy oversaturation.

As stated previously, the specific objective used in these examples of application of the Markov Adaptive Control algorithm was not specifically to minimize delay, but rather to minimize the queues on the intersection approaches; the delay performance characteristics presented above were an ancillary outcome of the specific objective. Relative to performance related to that specific objective, Fig. 14 presents representative values of

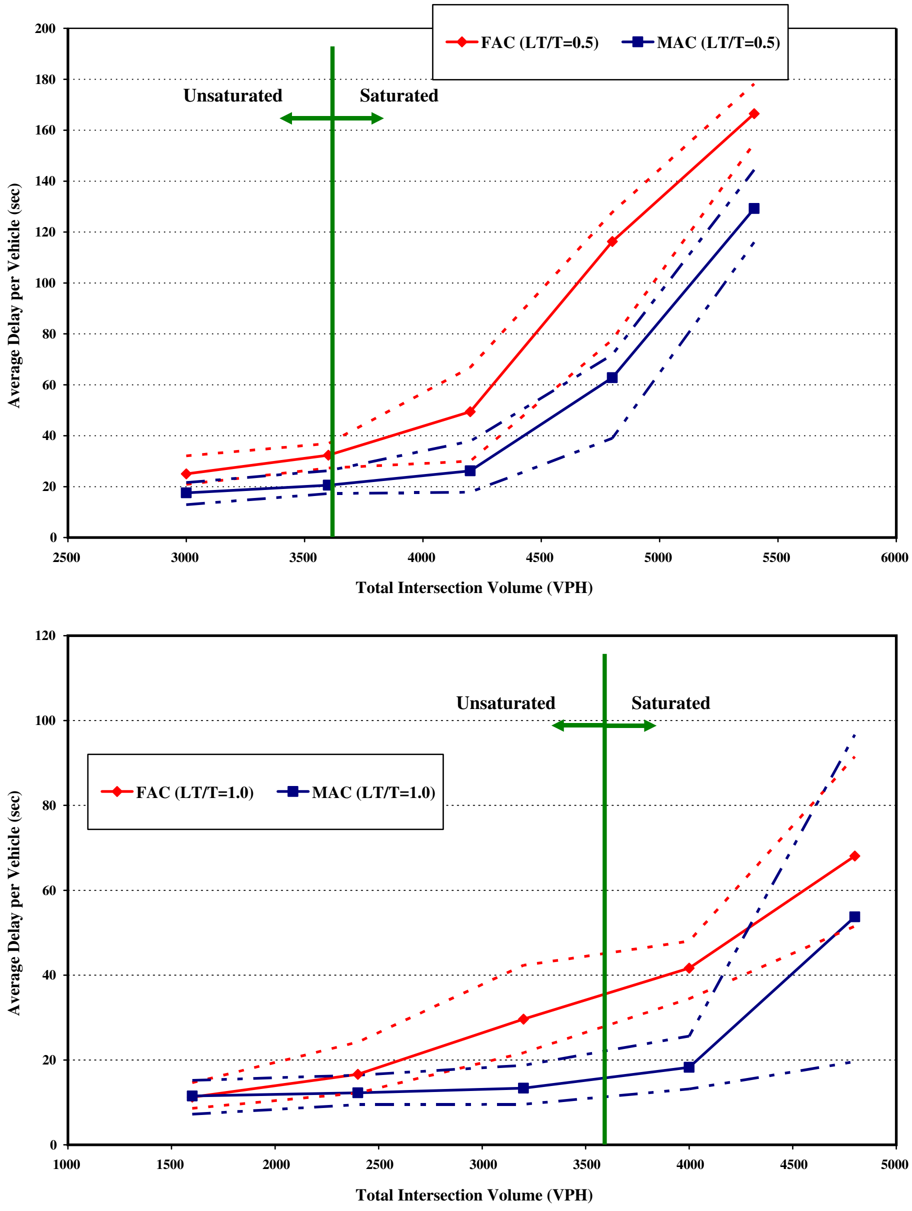

Fig. 11. Upper and lower bounds on simulation results.

the maximum queues for each movement obtained for the network case in which the total intersection volume is 3200 vph, or about 90% of intersection capacity.

The results indicate that the Markov Adaptive Control algorithm significantly outperforms full-actuated control in this aspect, although it must be noted that full-actuated control is not explicitly designed to minimize queue length, but rather implicitly works toward this end via its extension settings.

6. Summary and conclusions

Traffic signal control is a major ATMS component and its enhancement arguably is the most efficient way to reduce surface street congestion. The objective of the research presented here has been to present a more effective systematic approach to achieve real-time adaptive signal control for traffic networks.

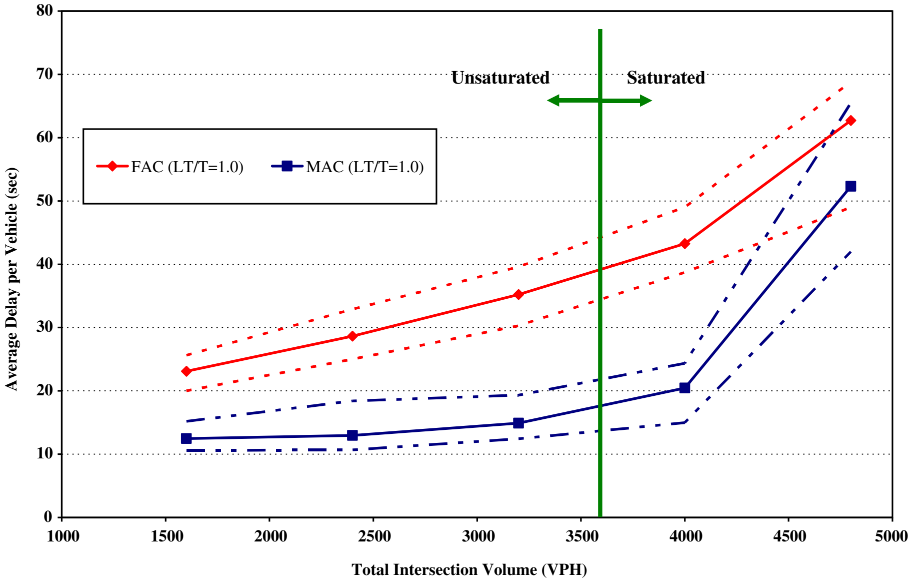

Fig. 12. Algorithm performance comparison for simple network case.

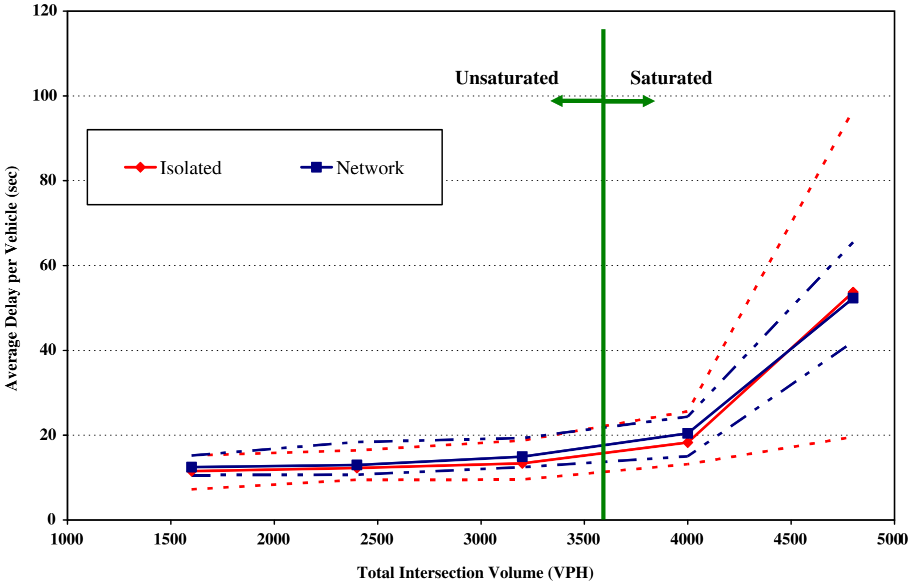

Fig. 13. MAC performance comparison between network and isolated intersection examples.

In this research, the problem of finding optimal traffic signal timing plans has been solved as a decision-making problem for a controlled Markov process. Controlled Markov processes have been used extensively to analyze and control complicated stochastic dynamical systems; its probabilistic, decision-making features match almost perfectly with the design features of a traffic signal control system. The Markovian model developed herein as the system model for signal control incorporates Robertson's platoon dispersion traffic model between intersections and employs the value iteration algorithm to find the optimal decision for the controlled Markov process. Analysis of computer simulation results indicates that this systematic approach is more efficient than the traditional full-actuated control, especially under the conditions of high traffic demand.

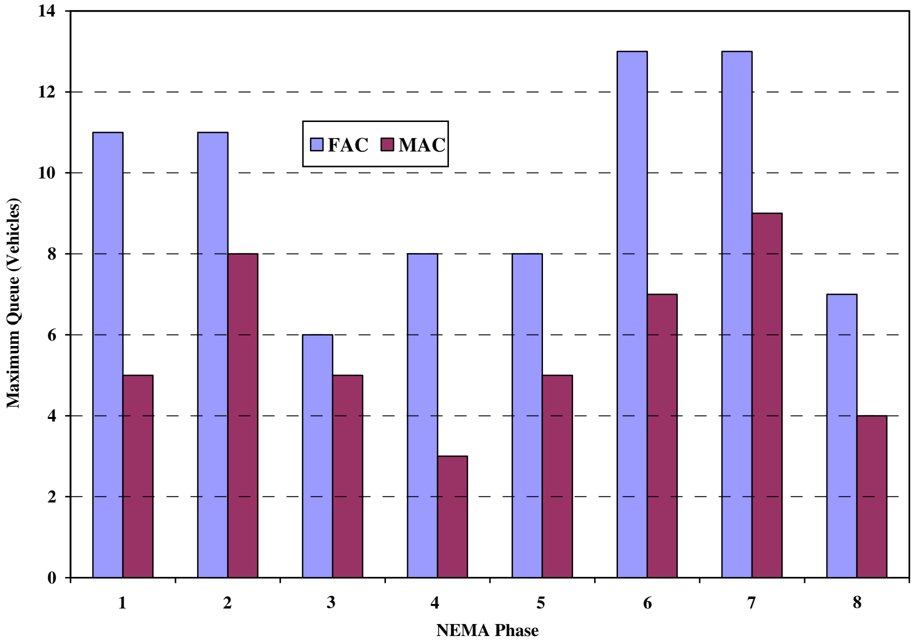

Fig. 14. Maximum queue comparison (3200 vph).

There are, of course, significant limitations to the present approach. Most notable is that as the size of the traffic network increases, i.e., the number of nodes/intersections and/or links increases, the dimension of the Markovian control model increases dramatically, requiring more memory space and computation time. This dimensionality issue is very important to real-time implementation, where processing speed is crucial. In the current formulation, one potential solution to this problem is alluded to by decomposing the network into sets of inter-linked network kernels of five intersections that could be handled by distributed/parallel processing protocols; however, no attempt has been made to thoroughly investigate the issues of such decomposition algorithms. Further, before any attempt to implement the results, a comprehensive sensitivity analysis needs to be conducted to study the effect of the various parameters employed in the simulation testing on both the performance of the model as well as on the objective function. Finally, for field testing, the original C language code must first be rewritten into assembly language; then the firmware can be loaded, or ''burned in'' to the PROM (Programmable Read-Only Memory) chip of the controller.

Appendix A. Successive approximation method—value iteration algorithm

The successive approximation algorithm given by Eq. (6) can be proved by Banach's fixed point theorem.

Banach's fixed point theorem states that if T is a contraction operator mapping a complete metric space (S, d) into itself, then T has a unique fixed point v^* such that for any $v \in S$ and $n \geqslant 0$:

$$d(T^n v, v^*) \leqslant \beta^n d(v, v^*), \qquad \text{(A.1)}\leftarrow$$

where $v^* \in S$ satisfies $Tv^* = v^*$, d is the metric and $0 \leqslant \beta < 1$.

To prove the operator T defined in Eq. (4) for the Markov process is a contraction operator, let $B(X)$ be the Banach space (a complete normed linear space) of real-valued bounded measurable functions on a Borel space X with supremum norm:

$$\|v\| = \sup_x |v|. \qquad \text{(A.2)}\leftarrow$$

For any $u \in B(X)$ and $v \in B(X)$, we have

$$|Tv(x) - Tu(x)| = \left| \max_{a\in A}\left[q(x,a) + \beta\sum_{j=1}^{N} v(x)p_{ij}^{a}\right] - \max_{a\in A}\left[q(x,a) + \beta\sum_{j=1}^{N} u(x)p_{ij}^{a}\right]\right|$$

$$\leqslant \beta \max_{a\in A}\left|\sum_{j=1}^{N}[v(x) - u(x)]p_{ij}^{a}\right| \leqslant \beta\|v(x) - u(x)\|.$$

Taking the supremum over all $x \in X$:

$$\|Tv - Tu\| \leqslant \beta\|v - u\|, \qquad \text{(A.3)}\leftarrow$$

where $\|\cdot\|$ is the supremum norm. Thus by the definition, T is a contraction operator.

Rewrite Eq. (6) as

$$v_n = Tv_{n-1} = T^n v_0$$

with arbitrary initial condition $v_0 \in V$, then by the Banach's fixed point theorem,

$$\|v_n - v^*\| = \|T^n v_0 - v^*\| \leqslant \beta^n\|v_0 - v^*\|. \qquad \text{(A.4)}\leftarrow$$

Since $0 \leqslant \beta \leqslant 1$ and v is bounded,

$$\operatorname*{limit}_{n\to\infty\leftarrow} v_n \to v^* \qquad \text{(A.5)}\leftarrow$$

i.e., the optimal reward, or the solution of Eq. (4), exists and can be approximated by the value-iteration algorithm. The uniqueness of this solution is also guaranteed by the Banach's fixed point theorem.

References

Chang, G.L., Tao, X., 1997. Estimation of time-dependent turning fractions at signalized intersections. Transportation Research Record 1644, 142–149.

Chang, G.-L., Wu, J., Lieu, H., 1994. Real-time incident-responsive corridor control: a successive linear programming approach. In: Proceedings of the Fourth Annual Meeting of IVHS America, Atlanta, GA, vol. 2, pp. 907–918.

Chen, A., Chootinan, P., Recker, W., 2005. Examining the quality of synthetic origin–destination trip table estimated by path flow estimator. Journal of Transportation Engineering, American Society of Civil Engineers 131 (7), 506–513.

Cremer, M., Schoof, S., 1989. On control strategies for urban traffic corridors. In: Proceedings of the XXX IFAC Conference on Control Computers and Communications in Transportation, Paris, France, pp. 213–219.

D'Ans, G.C., Gazis, D.C., 1976. Optimal control of oversaturated store-and forward transportation networks. Transportation Science 10, 1–19.

Davis, G.A., Lan, C.J., 1995. Estimating intersection turning movement proportions from less-than-complete sets of traffic counts. Transportation Research Record 1510, 53–59.

Gartner, N., 1983. OPAC: A demand-responsive strategy for traffic signal control. Transportation Research Record, 906.

Hakimi, S.L., 1969. Analysis and design of communication networks with memory. Journal of the Franklin Institute 287 (1), 1–17.

Head, K.L., 1995. An event-based short-term traffic flow prediction model. Transportation Research Record 1510, 45–52.

Hernandez-Lerma, O., 1989. Adaptive Markov Control Process. Applied Mathematical Sciences, vol. 79. Springer-Verlag.

Hunt, P.B., Robertson, D.I., et al., 1982. The SCOOT on-line traffic signal optimization technique. Traffic Engineering and Control. April.

Lin, F.B., 1989. Use of binary choice decision process for adaptive signal control. Journal of Transportation Engineering.

Lin, F.B., Vijayakumar, S., 1989. Adaptive signal control at isolated intersections. Journal of Transportation Engineering.

Lo, H.K., 2001. A cell-based traffic control formulation: strategies and benefits of dynamic timing plans. Transportation Science 35 (2), 148–164.

Lowrie, P., 1982. The Sydney coordinated adaptive control system—principles, methodology, algorithms. In: IEE Conference Publication, 207.

Maher, M.J., 1984. Estimating the turning flows at a junction: a comparison of three models. Traffic Engineering and Control 25, 19–22.

Mirchandani, P.B., Nobe, S., Wu, W., 2001. The use of on-line turning proportion estimation in real-time traffic-adaptive signal control. Transportation Research Record 1728, 80–86.

Nakatsuji, T., Kaku, T., 1991. Development of a self-organizing traffic control system using neural network models. Transportation Research Record 1324.

Papoulis, A., 1984. Brownian Movement and Markoff Processes. Probability, Random Variables, and Stochastic Processes, second ed. McGraw-Hill, New York, pp. 515–553.

Robertson, D.I., 1969. TRANSYT: A traffic network study tool. RRL Report LR 253, Road Research Laboratory, England.

Robertson, D.I., Bretherton, R.D., 1991. Optimizing networks of traffic signals in real-time: the SCOOT method. IEEE Transactions on Vehicular Technology 40, 1.

Singh, M.G., Tamura, H., 1974. Modelling and hierarchical optimization for oversaturated urban road traffic networks. International Journal of Control 20 (6), 913–934.

Stephanedes, Y.J., Chang, K.-K., 1993. Optimal control of freeway corridors. Journal of Transportation Engineering 119 (4), 504–514.

Wilshire, R., Black, R., et al., 1985. Traffic Control Systems Handbook, FHWA-IP-85-12.

Yu, Nie, Zhang, H.M., Recker, W.W., 2005. Inferring origin–destination trip matrices with a decoupled GLS path flow estimator. Transportation Research Part B: Methodological 39 (6), 497–518.