Quasi-dynamic Traffic Light Control for a Single Intersection

Yanfeng Geng and Christos G. Cassandras* Division of Systems Engineering and Center for Information and Systems Engineering Boston University, gengyf@bu.edu, cgc@bu.edu

Abstract

We address the traffic light control problem for a single intersection by viewing it as a stochastic hybrid system and developing a Stochastic Flow Model (SFM) for it. We adopt a quasi-dynamic control policy based on partial state information defined by detecting whether vehicle backlog is above or below a certain threshold, without the need to observe an exact vehicle count. The policy is parameterized by green and red cycle lengths which depend on this partial state information. Using Infinitesimal Perturbation Analysis (IPA), we derive on-line gradient estimators of an average traffic congestion metric with respect to these controllable green and red cycle lengths when the vehicle backlog is above or below the threshold. The estimators are used to iteratively adjust light cycle lengths so as to improve performance and, in conjunction with a standard gradient-based algorithm, to seek optimal values which adapt to changing traffic conditions. Simulation results are included to illustrate the approach and quantify the benefits of quasi-dynamic traffic light control over earlier static approaches.

1. Introduction

The Traffic Light Control (TLC) problem aims at dynamically controlling the flow of traffic at an intersection through the timing of green/red light cycles with the objective of reducing congestion, hence also the delays incurred by drivers. The more general problem involves a set of intersections and traffic lights with the objective of reducing overall congestion over an area covering multiple urban blocks. Control strategies employed for TLC problems are generally classified into two categories: *fixed-cycle strategies* and *traffic*- responsive strategies. Fixed-cycle Strategies are derived off-line based on historical constant demands and turning rates for each stream; traffic-responsive strategies make use of real-time measurements to calculate in real time the best signal settings [1]. Recent technological developments involving better, inexpensive sensors and wireless sensor networks have enabled the collection of data (e.g., counting vehicles in a specific road section) which can be used for traffic-responsive strategies. Thus, methodologies that would not be possible to implement not long ago are now becoming feasible. The approach proposed in this paper to the TLC problem is specifically intended to exploit these recent developments.

Numerous algorithms have been proposed to solve the TLC problem. It is formulated as a Mixed Integer Linear Programming (MILP) problem in [2], and as an Extended Linear Complementary Problem (ELCP) in [3]. A Markov Decision Process (MDP) approach has been proposed in [4] and Reinforcement Learning (RL) was used in [5]. A game theoretic viewpoint is given in [6]. Due to its complexity when viewed as an optimization problem, fuzzy logic is often used in both a single (isolated) junction [7] and multiple junctions [8]. The authors in [9] proposed an ALLONS-D framework (Adaptive Limited Lookahead Optimization of Network Signals - Decentralized), which is a decentralized method based on the Rolling Horizon (RH) concept. Perturbation analysis techniques were used in [10] and a formal approach using Infinitesimal Perturbation Analysis (IPA) to solve the TLC problem was presented in [11] for a single intersection.

In [12], we studied the TLC problem for a single intersection using a Stochastic Flow Model (SFM) and Infinitesimal Perturbation Analysis (IPA), and extended the method to multiple intersections in [13]. In this prior work, the green/red cycle lengths are viewed as controllable parameters. The traffic light controller adjusts their values based on data collected over an interval at the end of which an IPA estimator dictates the adjustments. The data consist of counters and timers for

^{*}The authors' work is supported in part by NSF under Grants EFRI-0735974 and CNS-1239021, by AFOSR under grant FA9550-12-1-0113, by ONR under grant N00014-09-1-1051, and by ARO under Grant W911NF-11-1-0227.

simple, directly observable events, but no state information (in the form of instantaneous vehicle backlogs) is used. In this paper, we make a first attempt to use state feedback for the controller in between two light cycle adjustment points and use this information to improve the adjustments made. However, since it is unrealistic to obtain instantaneous vehicle backlog information and derive a fully dynamic controller, we make use of partial state information and derive a *quasi-dynamic* controller. In particular, we define for each traffic flow *i* a minimum and maximum green light cycle length, $\theta_{i,1}$, $\theta_{i,2}$ respectively, and allow light switches as long as the cycle has exceeded $\theta_{i,1}$ depending *only* on whether some threshold of the vehicle backlog is reached, assuming that such events are observable. We use IPA to estimate the sensitivities of an average traffic congestion metric with respect to these parameters, hence improving and seeking to optimize overall performance.

In our analysis, we still adopt a stochastic hybrid system modeling framework [14],[15], since the problem involves both event-driven dynamics in the switching of traffic lights and time-driven dynamics that capture the flow of vehicles through an intersection. A SFM as introduced in [16] treats flow models as stochastic processes. In the TLC problem, this is consistent with continuously and randomly varying traffic flows, especially in heavy traffic conditions. With only minor technical assumptions imposed on the properties of such processes, a general IPA theory for stochastic hybrid systems was recently presented in [17],[18] through which one can estimate on line gradients of certain performance measures with respect to various controllable parameters. These estimates may be incorporated in standard gradient-based algorithms to optimize system parameter settings. IPA estimates become biased when dealing with aspects of queueing systems such as multiple user classes, blocking due to limited resource capacities, and various forms of feedback control. The use of IPA in stochastic hybrid systems, however, circumvents these limitations and yields simple unbiased gradient estimates (under mild technical conditions) of useful metrics (see [18].) We emphasize that the IPA gradient estimates we derive in the TLC are independent of the stochastic characteristics of all the vehicle traffic flows involved, rendering them robust to traffic variations and requiring no explicit models for the traffic flows.

The rest of this paper is organized as follows. In Section 2, we formulate the TLC problem for two intersections and construct a SFM. In Section III, we derive an IPA estimator for a cost function gradient with respect to a controllable parameter vector defined by green and red cycle lengths. Simulation-based examples are given in Section IV and we conclude with Section V.

2. Problem Formulation

A single isolated intersection is shown in Fig. 1, where there are two roads and two traffic lights, with each traffic light controlling the associated incoming traffic flow. For simplicity, we make the following assumptions: (*i*) Left-turn and right-turn traffic flows are not considered, i.e., traffic lights only control vehicles going straight. (*ii*) A YELLOW light is combined with a RED light (therefore, the YELLOW light duration is not explicitly controlled).



Figure 1: A single traffic intersection

2.1. Quasi-dynamic Control

We assign to each queue *i* a guaranteed minimum GREEN cycle length $\theta_{i,1}$, and a maximum length $\theta_{i,2}$. These are both controllable parameters so that the controllable parameter vector of interest is $\theta =$ $[\theta_{1,1}, \theta_{1,2}, \theta_{2,1}, \theta_{2,2}]$. We define a state vector $x(\theta, t) =$ $[x_1(\theta, t), x_2(\theta, t)]$ where $x_i(\theta, t) \in \mathbb{R}^+$ is the content of queue *i*. For each queue *i*, we also define a leftcontinuous "clock" state variable $z_i(\theta, t), i = 1, 2$, which measures the time since the last switch from RED to GREEN of the traffic light for queue *i*, therefore, $z_i(\theta, t) \in [0, \theta_{i,2}]$. Setting $z(\theta, t) = [z_1(\theta, t), z_2(\theta, t)]$, the complete system state vector is $[x(\theta, t), z(\theta, t)]$. For notational simplicity, we will write $x_i(t), z_i(t)$ when no confusion arises.

At any time *t*, the feasible control set for the traffic light controller is $U = \{1, 2\}$, where we define the control as

$$u(x(t), z(t)) \equiv \begin{cases} 1 & \text{set road 1 GREEN, road 2 RED} \\ 2 & \text{set road 2 GREEN, road 1 RED} \end{cases}$$
(1)

A *dynamic* controller is one that makes full use of the state information z(t) and x(t). Obviously, z(t) is the controller's known internal state, but the queue content state is generally not observable. We assume, however,

that it is partially observable. Specifically, we can only observe whether $x_i(t)$ is below or above some given threshold S_i , i = 1, 2 (this is consistent with actual traffic systems where a single sensor (typically, an inductive loop detector) is installed at each road near the intersection). Based on such partially observed states, we partition the queue content state space into four regions as follows:

$$\begin{aligned} X_0 &= \{ (x_1, x_2) : x_1(t) < S_1, \ x_2(t) < S_2 \} \\ X_1 &= \{ (x_1, x_2) : x_1(t) < S_1, \ x_2(t) \ge S_2 \} \\ X_2 &= \{ (x_1, x_2) : x_1(t) \ge S_1, \ x_2(t) < S_2 \} \\ X_3 &= \{ (x_1, x_2) : x_1(t) \ge S_1, \ x_2(t) \ge S_2 \} \end{aligned}$$

The *quasi-dynamic* controller we consider is of the form u(z(t), X(t)), where $X(t) = X_0, \ldots, X_3$, and is defined as follows. If $X(t) \in \{X_0, X_3\}$,

$$u(z(t)) = \begin{cases} 1 & \text{if } z_1(t) \in (0, \theta_{1,2}), z_2(t) = 0\\ 2 & \text{otherwise} \end{cases}$$
(2)

If $X(t) = X_1$,

$$u(z(t)) = \begin{cases} 1 & \text{if } z_1(t) \in (0, \theta_{1,1}), z_2(t) = 0\\ 2 & \text{otherwise} \end{cases}$$
(3)

If $X(t) = X_2$,

$$u(z(t)) = \begin{cases} 2 & \text{if } z_2(t) \in (0, \theta_{2,1}), z_1(t) = 0\\ 1 & \text{otherwise} \end{cases}$$
(4)

This is a simple form of hysteresis control satisfying the following simple rules with $j \neq i$:

- If the GREEN light cycle at queue *i* reaches θ_{i,2}, it switches to RED.
- If the GREEN light cycle at queue *i* reaches $\theta_{i,1}$, and $x_i < S_i$ (low *i* traffic), $x_j \ge S_j$ (high *j* traffic), it switches to RED.
- If the GREEN light cycle at queue *i* has exceeded θ_{i,1}, and x_i decreases below S_i but x_j ≥ S_j, it switches to RED.
- If the GREEN light cycle at queue *i* has exceeded θ_{i,1}, and x_j increases above S_j but x_i < S_i, it switches to RED.

Clearly, the GREEN light cycle may now be dynamically interrupted anytime after $\theta_{i,1}$ based on the partial state feedback provided through X(t). Let τ_k be the *k*th time instant when a GREEN cycle starts at queue *i* and let $j \neq i$. Define

$$r_i(\tau_k) = \min\{t | t \ge \tau_k, z_i(t) \ge \theta_{i1}, x_i(t) < S_i, x_j(t) \ge S_j\}$$

This is the earliest time when all conditions are satisfied to interrupt a GREEN cycle: the minimum GREEN cycle length θ_{i1} has been reached, the queue *i* length is low, and the competing queue *j* length is high. At $t = r_i(\tau_k)$, a GREEN to RED light switching event is forced as long as the residual GREEN cycle satisfies $\theta_{i,2} - r_i(\tau_k) > 0$. Therefore, the condition for such an interruption event is

$$r_i(\tau_k) < \theta_{i,2}$$

At $t = \theta_{i,2}$, the GREEN light is forced to switch to RED. This is an essential component of the quasi-dynamic controller we have defined. It will also be reflected by the state dynamics in the next subsection.

2.2. System Dynamics

The system, as described above, involves a number of stochastic processes which are all defined on a common probability space (Ω, F, P) . Each of the two roads is considered as a queue with a random *arrival* flow process $\{\alpha_n(t)\}, n = 1, 2$, where $\alpha_n(t)$ is the instantaneous vehicle arrival rate at time t. When the traffic light corresponding to road n is GREEN, the *departure* flow process is denoted by $\{\beta_n(t)\}, n = 1, 2$. We emphasize again that the IPA estimators we will derive do not require any knowledge of the processes $\{\alpha_n(t)\}$ and $\{\beta_n(t)\}$; only estimates of arrival and departure flows at specific observable event times are involved. As in prior work [18],[12], we assume only that $\alpha_n(t)$ and $\beta_n(t)$ are piecewise continuous w.p. 1.

We can now write the dynamics of each state variable $z_i(t)$ based on the control policy (1)-(4) as follows:

$$\dot{z}_i(t) = \begin{cases} 1 & \text{if } z_j(t) = 0, \ j \neq i \\ 0 & \text{otherwise} \end{cases}$$
(5)

$$z_i(t^+) = 0 \text{ if } z_i(t) = \theta_{i,2}$$
or
$$z_i(t) = \theta_{i,1}, x_i(t) < S_i, x_j(t) \ge S_j$$
(6)

or
$$z_i(t) > \theta_{i,1}, x_i(t^-) > S_i, x_i(t^+) = S_i, x_j(t) \ge S_j$$

or $z_i(t) > \theta_{i,1}, x_i(t) < S_i, x_j(t^-) < S_j, x_j(t^+) = S_j$

Note that $z_i(t)$ is reset to 0 as soon as a GREEN light switches to RED and it remains at this value while the light is GREEN for queue $j \neq i$.

The dynamics of each state variable $x_n(t)$ are as follows:

$$\dot{x}_n(t) = \begin{cases} \alpha_n(t) & \text{if } z_n(t) = 0\\ 0 & \text{if } x_n(t) = 0 \text{ and } \alpha_n(t) \le \beta_n(t)\\ \alpha_n(t) - \beta_n(t) & \text{otherwise} \end{cases}$$
(7)

Using the standard definition of a Stochastic Hybrid Automaton (SHA) (e.g., see [14]), we have a SHA

for the system as shown in Fig. 2. To simplify the automaton, we omit the dynamics of $x_n(t)$ and $z_n(t)$ and aggregate the states $x_n(t) = 0$ and $x_n(t) > 0$ as one state. As we can see, the system has 14 modes, which are defined by different combinations of $x_n(t)$ and $z_n(t)$. Two transient modes are not shown in the SHA: $z_1 \ge \theta_{1,1}, z_2 = 0, x_1 < S_1, x_2 \ge S_2$ and $z_1 = 0, z_2 \ge \theta_{2,1}, x_1 \ge S_1, x_2 < S_2$. It is easy to show that the control policies (3)-(4) force irreversible transitions into recurrent states in Fig. 2.



Figure 2: Stochastic Hybrid Automaton under Quasidynamic Control

Next, we define the set of all events in this system (causing all mode transitions) as $\Phi_n =$ $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$ where e_1 is the guard condition $[x_n = S_n$ from below]; e_2 is the guard condition $[x_n = S_n$ from above]; e_3 is the guard condition $[z_i = \theta_{i,1}]$, i.e., the GREEN cycle length reaches its lower limit; e_4 is the guard condition $[z_i = \theta_{i,2}]$, i.e., the GREEN cycle length reaches its upper limit; e_5 is the guard condition $[x_n = 0$ from above], i.e., the *n*th queue becomes empty; e_6 is a switch in the sign of $\alpha_n(t) - \beta_n(t)$ from non-positive to strictly positive; and e_7 is a switch in the sign of $\alpha_n(t)$ from 0 to strictly positive. Events e_5 , e_6 and e_7 are not shown in the above automaton (see [12] for detailed descriptions).

A typical sample path of any one of the queue content states (as shown in Fig. 3) consists of intervals over which $x_n(t) > 0$, which we call *Non-Empty Periods* (NEPs), followed by intervals where $x_n(t) = 0$, which we call *Empty Periods* (EPs). Thus, the entire sample path consists of a series of alternating NEPs and EPs. For easier reference, we let "E" denote any "NEP end" event (caused by e_5), "R2G" denote a light switching event from RED to GREEN, "G2R" denote a light switching event from GREEN to RED (both G2R and R2G are caused by e_1, \ldots, e_4), and "S" denote any "NEP start" event, which is caused by e_6 , e_7 or G2R (see also [12]). Our analysis will be based on studying these four event types.

In Fig. 3, the *m*th NEP in a sample path of any queue, m = 1, 2, ..., is denoted by $[\xi_{n,m}, \eta_{n,m})$, i.e., $\xi_{n,m}$, $\eta_{n,m}$ are the occurrence times of the *m*th S and E event respectively at this queue. During the *m*th NEP, $t_{n,m}^j$, $j = 1, ..., J_m$, denotes the time when a traffic light switching event occurs (either R2G or G2R).



Figure 3: A typical sample path of a traffic light queue

2.3. Objective Function

Our objective is to select θ so as to minimize a cost function that measures a weighted mean of the queue lengths over a fixed time interval [0,T]. Note that the threshold parameters S_i , i = 1, 2, are assumed to be fixed for the purpose of this paper. In particular, we define the sample function

$$L(\theta; x(0), z(0), T) = \frac{1}{T} \sum_{n=1}^{2} \int_{0}^{T} w_{n} x_{n}(\theta, t) dt \quad (8)$$

where w_n is a cost weight associated with queue n and x(0), z(0) are given initial conditions. It is obvious that since $x_n(t) = 0$ during EPs of queue n, we can rewrite (8) in the form

$$L(\theta; x(0), z(0), T) = \frac{1}{T} \sum_{n=1}^{2} \sum_{m=1}^{M_n} \int_{\xi_{n,m}}^{\eta_{n,m}} w_n x_n(\theta, t) dt$$
(9)

where M_n is the total number of NEPs during the sample path of queue n. For convenience, we also define

$$L_{n,m}(\boldsymbol{\theta}) = \int_{\xi_{n,m}}^{\eta_{n,m}} x_n(\boldsymbol{\theta}, t) dt$$
(10)

to be the sample cost associated with the *m*th NEP of queue n. We can now define our overall performance metric as

$$J(\theta; x(0), z(0), T) = E[L(\theta; x(0), z(0), T]$$
(11)

Recall that we do not impose any limitations on the processes $\{\alpha_n(t)\}$ and $\{\beta_n(t)\}$ and only assume that $\alpha_n(t)$, $\beta_n(t)$ are piecewise continuous w.p. 1. Thus,

it is infeasible to obtain a closed-form expression of $J(\theta; x(0), z(0), T)$. The value of IPA, as developed for general stochastic hybrid systems in [18], is in providing the means to estimate the performance metric gradient $\nabla J(\theta)$, by evaluating the sample gradient $\nabla L(\theta)$. As shown elsewhere (e.g., see [18]), these estimates are unbiased under mild technical conditions. Moreover, an important property of IPA estimates is that they are often independent of the unknown processes $\{\alpha_n(t)\}$ and $\{\beta_n(t)\}\$ or they depend on values of $\alpha_n(t)$ or $\beta_n(t)$ at specific event times only. Such robustness properties of IPA (formally established in [19]) make it attractive for estimating on line performance sensitivities with respect to controllable parameters such as θ in our case. One can then use this information to either improve performance or, under appropriate conditions, solve an optimization problem and determine an optimal θ^* through an iterative scheme:

$$\theta_{i,k+1} = \theta_{i,k} - \gamma_k H_{i,k}(\theta_k, x(0), T, \omega_k), k = 0, 1, \dots$$
 (12)

where $H_{i,k}(\theta_k, x(0), T, \omega_k)$ is an estimate of $dJ/d\theta_i$ based on the information obtained from the sample path denoted by ω_k , and γ_k is the stepsize at the *k*th iteration. Next we will focus on how to obtain $dL/d\theta$. We may then also obtain θ^* through (12), provided that $\{\alpha_n(t)\}$ and $\{\beta_n(t)\}$ are stationary.

3. Infinitesimal Perturbation Analysis (IPA)

To simplify notation,, we redefine $\theta_{1,1}$, $\theta_{1,2}$ as θ_1 , θ_2 and $\theta_{2,1}$, $\theta_{2,2}$ as θ_3 , θ_4 , and define the derivatives of the states $x_n(t, \theta)$ and $z_i(t, \theta)$ and event times $\tau_k(\theta)$ with respect to θ_i , i = 1, ..., 4, as follows:

$$x_{n,i}'(t) \equiv \frac{\partial x_n(\theta, t)}{\partial \theta_i}, \ z_{i,i}'(t, \theta) \equiv \frac{\partial z_i(\theta, t)}{\partial \theta_i}, \ \tau_{k,i}' \equiv \frac{\partial \tau_k(\theta)}{\partial \theta_i}$$
(13)

Consider a sample path of the system as modeled in Fig. 2 over [0, T] and let $\tau_k(\theta)$ denote the occurrence time of the *k*th event (of any type), where we stress its dependence on θ . Taking derivatives with respect to θ_i in (9), and observing that $x_n(\xi_{n,m}) = x_n(\eta_{n,m}) = 0$, we obtain

$$\frac{dL(\theta)}{d\theta_i} = \frac{1}{T} \sum_{n=1}^{2} \sum_{m=1}^{M_n} w_n \frac{dL_{n,m}(\theta)}{d\theta_i}$$
(14)

Observe that the determination of the sample derivatives depends on the state derivatives $x'_{n,i}(t)$. The purpose of IPA is to evaluate these derivatives as functions of observable sample path quantities. We pursue this next, using the framework established in [18]

where, for arbitrary stochastic hybrid systems, it is shown that the state and event time derivatives in (13) can be obtained from three fundamental "IPA equations". For the sake of self-sufficiency, these equations are rederived here as they pertain to our specific SFM.

3.1. IPA review

Looking at (7), note that the dynamics of $x_n(t)$ are fixed over any interevent interval $[\tau_k, \tau_{k+1})$ and we write $\dot{x}_n(t) = f_{n,k}(t)$ to represent the appropriate expression on the right-hand-side of (7) over this interval. We have $x_n(t) = x_n(\tau_k) + \int_{\tau_k}^t f_{n,k}(\tau) d\tau$. Taking derivatives with respect to θ_i and letting $t = \tau_k^+$, we obtain

$$x'_{n,i}(\tau_k^+) = x'_{n,i}(\tau_k^-) + [f_{n,k-1}(\tau_k^-) - f_{n,k}(\tau_k^+)] \cdot \tau'_{k,i}$$
(15)

Moreover, further taking derivatives with respect to *t*, we get, for all $t \in [\tau_k, \tau_{k+1})$,

$$\frac{d}{dt}x'_{n,i}(t) = \frac{\partial f_{n,k}}{\partial x_n}(t)x'_{n,i}(t) + \frac{\partial f_{n,k}}{\partial \theta_i}(t)$$
(16)

Since $\frac{\partial f_{n,k}}{\partial x_n} = \frac{\partial f_{n,k}}{\partial \theta_i} = 0$ and we get $\frac{d}{dt}x'_{n,i}(t) = 0$. Therefore, $x'_{n,i}(t)$ remains constant over all $t \in [\tau_k, \tau_{k+1})$:

$$x'_{n,i}(t) = x'_{n,i}(\tau_k^+), \quad t \in [\tau_k, \tau_{k+1})$$
 (17)

Thus, focusing on a NEP of $x_n(t)$, the queue content derivative is piecewise constant with jumps occurring according to (15). The next step is to obtain the event time derivatives $\tau'_{k,i}$ appearing in (15).

Clearly $\tau'_{k,i}$ depends on the type of event occurring at time τ_k . Following the framework in [18], there are three types of events for a general stochastic hybrid system. (*i*) *Exogenous Events*. These events cause a discrete state transition independent of θ and satisfy $\tau'_{k,i} = 0$. (*ii*) *Endogenous Events*. Such an event occurs at time τ_k if there exists a continuously differentiable function $g_k : \mathbb{R}^n \times \Theta \to \mathbb{R}$ such that $\tau_k = \min\{t > \tau_{k-1} : g_k(x(\theta,t), \theta) = 0\}$, where the function g_k normally corresponds to a guard condition in a hybrid automaton. Taking derivatives with respect to θ_i , i = 1, ..., m, it is straightforward to obtain

$$\tau_{k,i}' = -\frac{\frac{\partial g_k}{\partial \theta_i} + \sum_{j=1}^{N} \frac{\partial g_k}{\partial x_j} \chi_{j,i}'(\tau_k^-)}{\sum_{j=1}^{N} \frac{\partial g_k}{\partial x_j} f_{j,k-1}(\tau_k^-)}$$
(18)

(*iii*) *Induced Events*. Such an event occurs at time τ_k if it is triggered by the occurrence of another event at time $\tau_m \leq \tau_k$. (details can be found in [18].)

3.2. State and event time derivatives

In the following, we consider each of the four event types (E, S, R2G, G2R) for queue n that were identified in the previous section and derive the corresponding event time derivatives. Based on these, we can then also derive the state derivatives through (15) and (17).

(1) Event *E* ending a NEP. This is an endogenous event that occurs when $x_n(\theta,t) = 0$. Thus, when such an event occurs at τ_k , let $g_k(x(\theta,t),\theta) = x_n(\theta,t) = 0$. Using (18), we get $\tau'_{k,i} = \frac{-x'_{n,i}(\tau_k^-)}{f_{n,k-1}(\tau_k^-)}$. Looking at (7), note that $f_{n,k-1}(\tau_k^-) = \alpha_n(\tau_k) - \beta_n(\tau_k)$ and $f_{n,k}(\tau_k^+) = 0$. Using these values in (15) along with $\tau'_{k,i}$ above we get

$$x_{n,i}'(\tau_k^+) = x_{n,i}'(\tau_k^-) - \frac{[\alpha_n(\tau_k) - \beta_n(\tau_k)]x_{n,i}'(\tau_k^-)}{\alpha_n(\tau_k) - \beta_n(\tau_k)} = 0$$

Thus, at the end of a NEP $[\xi_{n,m}, \eta_{n,m})$ of queue *n* we have

$$x'_{n,i}(\eta^+_{n,m}) = 0 \tag{19}$$

indicating that these state derivatives are always reset to 0 upon ending a NEP.

(2) G2R event. If a G2R event occurs within a NEP (i.e., $x_n(\tau_k) > 0$), then, based on (7), we have $f_{n,k-1}(\tau_k^-) = \alpha_n(\tau_k) - \beta_n(\tau_k)$ and $f_{n,k}(\tau_k^+) = \alpha_n(\tau_k)$. Therefore, from (15) we get $x'_{n,i}(\tau_k^+) = x'_{n,i}(\tau_k^-) - \beta_n(\tau_k) \cdot \tau'_{k,i}$. If, on the other hand, $x_n(\tau_k) = 0$, then $f_{n,k-1}(\tau_k^-) = 0$ and we get $x'_{n,i}(\tau_k^+) = x'_{n,i}(\tau_k^-) - \alpha_n(\tau_k) \cdot \tau'_{k,i}$. Combining these two results,

$$x_{n,i}'(\tau_k^+) = x_{n,i}'(\tau_k^-) - \begin{cases} \beta_n(\tau_k) \cdot \tau_{k,i}' & \text{if } x_n(\tau_k) > 0\\ \alpha_n(\tau_k) \cdot \tau_{k,i}' & \text{if } x_n(\tau_k) = 0 \end{cases}$$
(20)

Next we consider all four events e_1, \ldots, e_4 which may trigger the G2R event.

Case (2a): G2R event at queue *i* is triggered by $z_i = \theta_{i,2}$ (e_4 event). This is an endogenous event. In the model studied in [12], G2R and R2G events alternate with fixed GREEN/RED cycles and $\tau'_{k,i}$ is obtained by simply counting traffic light switches. However, under the quasi-dynamic control considered here, the GREEN cycle length can take any value in $[\theta_{i,1}, \theta_{i,2}]$ and the same method no longer applies. In the following lemma we derive $\tau'_{k,i}$.

Lemma 1 Let τ_k be the time of a G2R event induced by $z_n = \theta_{n,2}, n = 1, 2$, and τ_p be the last R2G event time before τ_k . We then have:

$$\tau'_{k,i} = \begin{cases} 1 + \tau'_{p,i} & \text{if } i = 2, n = 1 \text{ or } i = 4, n = 2 \\ \tau'_{p,i} & \text{otherwise} \end{cases}$$
(21)

Proof: Without loss of generality, we use queue 1, i.e., n = 1 above, to derive $\tau'_{k,i} = \frac{\partial \tau_k(\theta)}{\partial \theta_i}$ for all i = 1, 2, 3, 4. Setting

$$g_k(x(\boldsymbol{\theta},t),\boldsymbol{\theta}) = z_1 - \boldsymbol{\theta}_2 = 0 \tag{22}$$

in (18) and taking derivatives with respect to θ_1 above, we have

$$\tau_{k,1}' = -\left(\dot{z}_1(\tau_k^-)\right)^{-1} \left(z_{1,1}'(\tau_k^-) - 0\right) = -z_{1,1}'(\tau_k^-) \quad (23)$$

where $\dot{z}_1(\tau_k^-)) = 1$ since

$$\dot{z}_1(t) = f_{1,k-1}(t) = \begin{cases} 1 & \text{if } z_2(t) = 0\\ 0 & \text{otherwise} \end{cases}$$

By the definition of τ_p , for any $t \in [\tau_p, \tau_k)$, $f_1(t) = \dot{z}_1(t) = 1$. Using (16), we get

$$\frac{d}{dt}z'_{1,1}(t) = 0 \tag{24}$$

so that $z'_{1,1}(t)$ remains constant for all $t \in [\tau_p, \tau_k)$, i.e.,

$$z_{1,1}'(\tau_k^-) = z_{1,1}'(\tau_p^+)$$

At time τ_p , according to (15), we have

$$z_{1,1}'(\tau_p^+) = z_{1,1}'(\tau_p^-) + [f_{1,p-1}(\tau_p^-) - f_{1,p}(\tau_p^+)] \cdot \tau_{p,1}'$$
(25)

Define τ_r to be the last G2R event time before τ_p . At τ_r , $z_1(\tau_r^+)$ is reset to 0 according to (5), so that $z'_{1,1}(\tau_r^+) = 0$. For any $t \in [\tau_r, \tau_p)$, we have $f_{1,p-1}(t) = \dot{z}_1(t) = 0$, thus $z'_{1,1}(\tau_p^-) = z'_{1,1}(\tau_r^+) = 0$. The light switches to GREEN after τ_p , so we have $f_{1,p}(\tau_p^+) = 1$. Hence, (25) becomes:

$$z_{1,1}^{'}(\tau_{p}^{+}) = -\tau_{p,1}^{'}$$

Returning to (23), we get

$$\tau'_{k,1} = -z'_{1,1}(\tau_k^-) = -z'_{1,1}(\tau_p^+) = \tau'_{p,1}$$
(26)

Following the same analysis, we have

$$\tau'_{k,3} = \tau'_{p,3}$$
 (27)

$$\tau'_{k,4} = \tau'_{p,4}$$
 (28)

This leaves only $\tau'_{k,2}$ to consider. Taking derivatives with respect to θ_2 in (22):

$$\tau_{k,2}' = -\left(\dot{z}_{1}(\tau_{k}^{-})\right)^{-1}\left(z_{1,2}'(\tau_{k}^{-}) - 1\right) = 1 - z_{1,2}'(\tau_{k}^{-})$$

As in (24), for any $t \in [\tau_p, \tau_k)$,

$$\frac{d}{dt}z_{1,2}'(t) = 0$$

and we get $z_{1,2}^{'}(\tau_k^-) = z_{1,2}^{'}(\tau_p^+)$. At time τ_p , we have

 $z_{1,2}^{'}(\tau_p^+) = z_{1,2}^{'}(\tau_p^-) + [f_{1,p-1}(\tau_p^-) - f_{1,p}(\tau_p^+)] \cdot \tau_{p,2}^{'} = -\tau_{p,2}^{'}$ and we finally get

$$\tau'_{k,2} = 1 - z'_{1,2}(\tau_k^-) = 1 + \tau'_{p,2}$$
(29)

This result leaves the value of $\tau'_{p,i}$ unspecified. In fact, this is the time when an R2G event occurs, which is **Case (3)** considered in the sequel where we shall derive explicit expressions for $\tau'_{p,i}$.

Case (2b): G2R event at queue *i* is triggered by $z_i = \theta_{i,1}$, where $x_i < S_i$ and $x_j \ge S_j$ (e_3 event). We use a similar lemma for this case.

Lemma 2 Let τ_k be the time of a G2R event induced by $z_n = \theta_{n,1}, n = 1, 2$, and τ_p be the last R2G event before τ_k . We then have

$$\tau'_{k,i} = \begin{cases} 1 + \tau'_{p,i} & \text{if } i = 1, n = 1 \text{ or } i = 3, n = 2\\ \tau'_{p,i} & \text{otherwise} \end{cases}$$
(30)

Proof: The proof is similar to that of Lemma 1 and is omitted.

Case (2c): G2R event at queue *i* is triggered by $x_i = S_i$ from above, where $z_i > \theta_{i,1}$ and $x_j \ge S_j$ (e_2 event). This is an endogenous event with $g_k(x(\theta,t),\theta) = x_n(\theta,t) - S_i = 0$. Using (18), with $f_{n,k-1}(\tau_k^-) = \alpha_n(\tau_k^-) - \beta_n(\tau_k^-)$ we get

$$\tau_{k,i}' = \frac{-x_{n,i}'(\tau_k^-)}{\alpha_n(\tau_k^-) - \beta_n(\tau_k^-)}$$
(31)

Case (2d): G2R event at queue *i* is triggered by $x_j = S_j$ from below, where $z_i > \theta_{i,1}$ and $x_i \le S_i$ (e_1 event). This is also an endogenous event with $g_k(x(\theta,t),\theta) = x_j(\theta,t) - S_j = 0$. Using (18), with $f_{j,k-1}(\tau_k^-) = \alpha_j(\tau_k^-)$ we get

$$\tau_{k,i}' = \frac{-x_{j,i}'(\tau_k^-)}{\alpha_j(\tau_k^-)} \tag{32}$$

(3) *R2G event*. If the R2G event occurs within a NEP (i.e., $x_n(\tau_p) > 0$), then, based on (7), we have $f_{n,p-1}(\tau_p^-) = \alpha_n(\tau_p)$ and $f_{n,p}(\tau_p^+) = \alpha_n(\tau_p) - \beta_n(\tau_p)$. From (15) we get

$$x'_{n,i}(\tau_p^+) = x'_{n,i}(\tau_p^-) + \beta_n(\tau_p) \cdot \tau'_{p,i}$$
(33)

The derivation of $\tau'_{p,i}$ is similar to **Case (2)** as detailed next.

Case (3a): R2G event at queue *i* is triggered by $g_p(x(\theta,t), \theta) = z_j - \theta_{j,2} = 0$ (e_3 event). This is an endogenous event. We have a lemma similar to Lemma 1 whose proof is therefore omitted:

Lemma 3 Let τ_p be the time of a R2G event induced by $z_j = \theta_{j,2}, j = 1, 2$, and τ_r be the last G2R event before τ_p . We then have

$$\tau'_{p,i} = \begin{cases} 1 + \tau'_{r,i} & \text{if } i = 2, j = 1 \text{ or } i = 4, j = 2\\ \tau'_{r,i} & \text{otherwise} \end{cases}$$
(34)

Case (3b): R2G event at queue *i* is triggered by $z_j = \theta_{j,1}$, where $x_j < S_j$ and $x_i \ge S_i$ (e_3 event). This is an endogenous event and we have another lemma similar to Lemma 1 whose proof is also omitted:

Lemma 4 Let τ_p be the time of a R2G event induced by $z_j = \theta_{j,1}, j = 1, 2$, and τ_r be the last G2R event time before τ_p . We then have

$$\tau'_{p,i} = \begin{cases} 1 + \tau'_{r,i} & \text{if } i = 1, j = 1 \text{ or } i = 3, j = 2 \\ \tau'_{r,i} & \text{otherwise} \end{cases}$$
(35)

Case (3c): R2G event at queue *i* is triggered by $x_j = S_j$ from above, where $z_j > \theta_{j,1}$ and $x_i \ge S_i$ (e_2 event). This is an endogenous event with $g_p(x(\theta,t),\theta) = x_j(\theta,t) - S_j = 0$. Using (18) with $f_{j,p-1}(\tau_p^-) = \alpha_j(\tau_p^-) - \beta_j(\tau_p^-)$, we get

$$\tau'_{p,i} = \frac{-x'_{j,i}(\tau_p^-)}{\alpha_j(\tau_p^-) - \beta_j(\tau_p^-)}$$
(36)

Case (3d): R2G event at queue *i* is triggered by $x_i = S_i$ from below, where $z_j > \theta_{j,1}$ and $x_j \le S_j$ (e_1 event). This is an endogenous event with $g_p(x(\theta,t),\theta) = x_n(\theta,t) - S_i = 0$. Using (18) with $f_{n,p-1}(\tau_p^-) = \alpha_n(\tau_p^-)$, we get

$$\tau_{p,i}' = \frac{-x_{n,i}'(\tau_p^-)}{\alpha_n(\tau_p^-)}$$
(37)

(4) *Event S starting a NEP*. There are three possible cases to consider as follows.

Case (4a): A NEP starts right after a G2R event. This is an endogenous event and was analyzed in Case (2) with $x_n(\tau_k) = 0$ in (20), i.e., $x'_{n,i}(\tau_k^+) = x'_{n,i}(\tau_k^-) - \alpha_n(\tau_k)\zeta_{n,k}$. Since $x'_{n,i}(\xi_{n,m}^-) = x'_{n,i}(\eta_{n,m-1}^+) = 0$, we get

$$x_{n,i}'(\tau_k^+) = -\alpha_n(\tau_k)\tau_{k,i}'$$
(38)

Case (4b): A NEP starts while $z_i = 0$, $z_j > 0$. This is an exogenous event occurring during a RED cycle for queue *n* and is due to a change in $\alpha_n(t)$ from a zero to a strictly positive value. Therefore, $\tau'_{k,i} = 0$, and

$$x'_{n,i}(\tau_k^+) = 0 \tag{39}$$

Case (4c): A NEP starts while $z_j = 0$, $z_i > 0$. This is an exogenous event occurring during a GREEN cycle for queue *n* due to a change in $\alpha_n(t)$ or $\beta_n(t)$ that results in $\alpha_n(t) - \beta_n(t)$ switching from a non-positive to a strictly positive value. The analysis is the same as **Case (4b)**.

This completes the derivation of all state and event time derivatives required to evaluate the sample performance derivative in (14).

3.3. Cost Derivative

Using the definition of $L_{n,m}(\theta)$ in (10), note that we can decompose (14) for each *n* into its NEPs and evaluate the derivatives $dL_{n,m}(\theta)/d\theta_i$. By virtue of (17), $x'_{n,i}(t)$ is piecewise constant during a NEP and its value changes only at an event point $t^j_{n,m}$, $j = 1, ..., J_{n,m}$. Therefore, we have

$$\frac{dL_{n,m}(\theta)}{d\theta_{i}} = x'_{n,i}((\xi_{n,m})^{+})(t^{1}_{n,m} - \xi_{n,m}) + x'_{n,i}((t^{J_{n,m}}_{n,m})^{+}) \cdot (\eta_{n,m} - t^{J_{n,m}}_{n,m}) + \sum_{j=2}^{J_{n,m}} x'_{n,i}((t^{j}_{n,m})^{+})(t^{j}_{n,m} - t^{j-1}_{n,m})$$

$$(40)$$

Clearly, $x'_{n,i}$ at each event time is determined by (15) which in turn depends on the event type at $t_{n,m}^{j}$, j = $1, ..., J_{n,m}$ and is given by the corresponding expression in (19), (20), (33), and (38)-(39). The associated event time derivatives in these expressions are provided recursively through (21), (30)-(32) and (34)-(37). Note that $dL_{n,m}(\theta)/d\theta_i$ can be computed using an on-line algorithm that updates $\tau'_{k,i}$ and $x'_{n,i}$ after every observed event. More importantly, this IPA derivative depends on: (i) the number of events in each NEP $J_{n,m}$, (ii) the number of $G2R_n$ events and $R2G_n$ events, (iii) the event times $\xi_{n,m}$, $\eta_{n,m}$ and $t_{n,m}^{j}$, and (iv) the arrival and departure rates $\alpha_n(\tau_k)$, $\beta_n(\tau_k)$ at an event time *only*. The quantities in (i) - (iii) are easily observed through counters and timers. The rates in (iv) may be obtained through simple rate estimators, emphasizing that they are only needed at a finite number of observed event times. As already pointed out, the detailed nature of $\alpha_n(t), \beta_n(t)$ is not required in (40).

4. Simulation Results

We describe how the IPA estimator derived for the SFM can be used to improve performance and ultimately determine optimal light cycles for an intersection modeled as a Discrete Event System (DES). We apply the IPA estimator using actual data from an ob-



Figure 4: Sample cost (J) and parameter (θ) trajectories

served sample path of this DES (in this case, by simulating it as a pure DES).

We assume vehicles arrive according to a Poisson process with rate $\bar{\alpha}_n$, n = 1, 2 (as already emphasized, our results are independent of this distribution.). We also assume vehicles depart at a fix rate β_n if the road is not empty. We constrain $\theta_{i,1}$, i = 1, 2, to take values in $[\theta_{\min,1}, \theta_{\max,1}]$, and constrain $\theta_{i,2}$, to take values in $[\theta_{i,1}, \theta_{\max,2}]$.

For the simulated DES model, we use a bruteforce (BF) method to find an optimal θ_{BF}^* : we discretize all real values of θ_i and for combinations of $\theta_i, i = 1, ..., 4$ (based on our previous definition, $\theta_1 \equiv$ $\theta_{1,1}, \theta_2 \equiv \theta_{1,2}, \theta_3 \equiv \theta_{2,1}, \theta_4 \equiv \theta_{2,2}$), we run 10 sample paths to obtain the average total cost. The value of θ_{BF}^* is the one generating the least average cost, to be compared to θ_{IPA}^* , the IPA-based method. In our simulations, we estimate $\alpha_n(\xi_{n,m})$ through N_a/t_w by counting vehicle arrivals N_a over a time window t_w before or after $\xi_{n,m}; \beta_n(\tau_k)$ is similarly estimated.

In all our simulations, we set $\theta_{\min,1} = 10sec$, $\theta_{\max,1} = 20sec, \ \theta_{\max,2} = 40sec, \ \beta_1 = \beta_2 = 1 \ \text{and} \ T =$ 2000sec. We also set the weight $w_i = 1$ if $x_i < S_i$, and $w_i = 10$ if $x_i \ge S_i$, i = 1, 2, which indicates there is more cost if the queue content exceeds the threshold. In Fig. 4, we show sample trajectories of J and θ where we set $1/\alpha = [1.9, 3]$ and S = [8, 8]. As we can see, the gradient-based algorithm converges quickly with an optimal cost $J_{IPA}^* = 51.68 < J_{BF}^* = 62.06$ obtained by the BF method. Extensive additional comparison results under different traffic intensities (denoted by α) are provided in Table 1. Generally, the IPA method gives similar performance with the BF method, which, however, becomes impractical when there are more controllable parameters, or when the ranges of the parameters are large.

Of greater interest is a comparison of the quasidynamic control to the one in [12], which uses a static controller based on θ^* in between two adjustment points. We use the same settings for the two models,

Table 1: IPA vs BF method with different α

$1/\bar{lpha}$	BE		ΙΡΔ	
	θ^*	J^{*}	θ	
[2.2,2.7]	[10,20,10,14]	12.7	[10,15.1,11.3,11.3]	
[2,3]	[11,30,10,14]	12.3	[10.2,19.3,10.1,16.3]	
[1.9,3]	[10,30,10,18]	16.4	[18.9,25.6,13.3,17.0]	
[1.8,3]	[14,30,10,20]	17.6	[10.1,20.0,10.1,12.1]	
[1.7,3]	[12,26,10,12]	24.6	[10.1,20.1,10.6,11.9]	

where we constrain $\theta \in [10, 40]$ in [12]. In Fig. 5 we show the cost trajectories of the two methods using the settings as in Fig. 4 and the same initial $\theta_0 = [20, 10]$. The quasi-dynamic IPA-based method converges to a lower cost. More comparison results are shown in Fig. 6 where the *x*-axis denotes the five traffic intensities in Table 1. As we can see, using quasi-dynamic control generally results in better performance than using static control, as expected.



Figure 5: Sample cost (J) trajectories using two static and quasi-dynamic control



Figure 6: Optimal cost comparisons under different traffic loads

In all simulations above, we use exponentially dis-

tributed interarrival time for the incoming traffic flow. Thus the traffic generally does not significantly vary due to the relatively low variance of the Poisson process. We expect to see more drastic improvement using our method under highly variable traffic. To justify this, we randomly add a burst of traffic into road lobserve the resulting "adaptivity" of our method. In Fig. 7 we compare the cost reduction under purely desponential traffic (exp) with the one under "disturbed" destrict (disturb). The y-axis denotes the percentage of cost reduction: $\frac{J_{fixed}^{*} - J_{IPA}^{*}}{J_{fixed}^{*}} * 100$, where J_{fixed}^{*} is the optimal value obtained using the IPA method in [12]. As we can see, quasi-dynamic control exhibits higher cost reduction under highly variable traffic.



Figure 7: Cost reduction comparisons under different traffic loads

5. Conclusions and Future work

We have developed a SFM for a single intersection traffic light control problem, based on which we derive an IPA gradient estimator of a cost metric with respect to the controllable green and red cycle lengths. The estimator is used to iteratively adjust light cycle lengths to improve performance and, under proper conditions, obtain optimal values which adapt to changing traffic conditions. In contrast to prior work [12], we apply quasi-dynamic cycle length control between adjustment points, using partial state information defined by detecting whether a vehicle backlog is above or below a certain threshold, without the need to observe an exact vehicle count. Numerical results show that the IPA method leads to better performance than that obtained through repetitive "brute-force" simulation, and better than using static cycle control. Future work will focus on using IPA to solve the TLC problem for an intersection with more complicated traffic (e.g., left-turn and right-turn), modeling accelerating traffic following a GREEN light and extending our method to solving the TLC problem over multiple junctions. Moreover, we will explore the use of IPA in controlling the queue content thresholds in addition to the light cycle lengths.

References

- M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043 – 2067, 2003.
- [2] Y. Dujardin, F. Boillot, D. Vanderpooten, and P. Vinant, "Multiobjective and multimodal adaptive traffic light control on single junctions," *14th IEEE Conference* on Intelligent Transportation Systems, pp. 1361–1368, 2011.
- [3] B. DeSchutter, "Optimal traffic light control for a single intersection," *Proceedings of the IEEE American Control Conference*, vol. 3, pp. 2195–2199, 1999.
- [4] X. Yu and W. Recker, "Stochastic adaptive control model for traffic signal systems," *Transportation Research Part C: Emerging Technology*, vol. 14, no. 4, pp. 263–282, 2006.
- [5] T. Thorpe, Vehicle traffic light control using sarsa. Master thesis, Dept. of Comp. Sci., Colorado State Univ., 1997.
- [6] I. Alvarez and A. Poznyak, "Game theory applied to urban traffic control problem," *Intertional Conference on Control, Automation and Systems*, pp. 2164–2169, 2010.
- [7] Y. S. Murat and E. Gedizlioglu, "A fuzzy logic multiphased signal control model for isolated junctions," *Transportation Research Part C*, vol. 18, pp. 19–36, 2005.
- [8] W. Choi, H. Yoon, K. Kim, I. Chung, and S. Lee, "A traffic light controlling flc considering the traffic congestion," *AFSS 2002, International Conference on Fuzzy Systems*, pp. 69–75, 2002.
- [9] I. Porche and S. Lafortune, "Adptive look-ahead optimization of traffic signals," *ITS Journal*, vol. 4, pp. 209– 254, 1999.
- [10] L. Head, F. Ciarallo, and D. L. an V. Kaduwela, "A perturbation analysis approach to traffic signal optimization," *INFORMS National Meeting*, 1996.
- [11] C. G. Panayiotou, W. C. Howell, and M. Fu, "Online traffic light control through gradient estimation using stochastic fluid models," *Proceedings of the IFAC 16th Triennial World Congress*, 2005.
- [12] Y. Geng and C. Cassandras, "Traffic light control using infinitesimal perturbation analysis," 51st IEEE Conference on Decision and Control, pp. 7001 – 7006, 2012.
- [13] Y. Geng and C. G. Cassandras, "Multi-intersection traffic light control using infinitesimal perturbation analysis," *11th International Workshop on Discrete Event Systems*, pp. 104 – 109, 2012.
- [14] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems, 2nd ed.* Springer, 2008.
- [15] C. G. Cassandras and J. Lygeros, *Stochastic Hybrid Systems*. Taylor and Francis, 2006.

- [16] C. G. Cassandras, Y. Wardi, B. Melamed, G. Sun, and C. G. Panayiotou, "Perturbation analysis for on-line control and optimization of stochastic fluid models," *IEEE Trans. Automat. Control*, vol. 47, no. 8, pp. 1234–1248, 2002.
- [17] Y. Wardi, R. Adams, and B. Melamed, "A unified approach to infinitesimal perturbation analysis in stochastic flow models: the single-stage case," *IEEE Trans. Automat. Control*, vol. 55, no. 1, pp. 89–103, 2010.
- [18] C. G. Cassandras, Y. Wardi, C. G. Panayiotou, and C. Yao, "Perturbation analysis and optimization of stochastic hybrid systems," *Europ. J. of Control*, vol. 16, no. 6, pp. 642–664, 2010.
- [19] C. Yao and C. Cassandras, "Perturbation analysis of stochastic hybrid systems and applications to resource contention games," *Frontiers of Electrical and Electronic Engineering in China*, vol. 6, no. 3, pp. 453–467, 2011.