Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, October 6-9, 2013

Estimating traffic control strategies with inverse optimal control

Sudeep Gowrishankar[1] and Daniel B. Work[2]

Abstract— **This article formulates the problem of estimating the traffic control strategy of a single intersection traffic signal controller from observed data. Building on previous results in the literature on optimal traffic signal control and convex inverse optimal control problems, a method is proposed to efficiently solve the inverse optimal traffic signal control problem to recover the control objective of the observed traffic controller. The resulting program is tested in a numerical experiment with synthetically generated data. Supporting source code is available for download at https://github.com/sgowris2/InverseOptimalControl.**

I. Introduction

A. Motivation

Urban traffic estimation and forecasting is a challenging problem, especially when knowledge of the traffic signal control policy is unknown to the estimator. Often, signal control strategies are known by the local authorities, but the information is difficult to obtain at larger scales. In other cases, the control strategies are based on proprietary algorithms, and thus cannot be obtained explicitly, even from the managing agencies.

During *extreme congestion events* such as sporting events and natural disasters, the traffic signal control is performed manually by traffic management police officers. Some reasons for manual control of traffic include the failure of traffic lights during disasters due to power outages, and the fact that the controller itself may not be optimized for extreme congestion events due to their rare occurrence. In each of the above situations, it is important to be able to quickly learn the control strategy of the controller (whether human or not) so that traffic prediction and estimation systems can integrate information on how the flow is regulated.

Therefore, in this article, we propose a method to address the following problem. *Is it possible to learn the control objective of a traffic signal, by observing the queue lengths at the intersection and the corresponding control actions?* This work is motivated by the development of *TrafficTurk* [19], an Android mobile application that allows for low–cost, real–time, and rapidly deployable temporary traffic sensing. This sensing platform is being developed to enable accurate real–time traffic estimation and prediction of urban traffic during extreme congestion events. Inspired by the 18th century fake chess playing machine known as the *mechanical turk* [18], and the classical turning movement counters used by transportation engineers for decades, *TrafficTurk* provides an intuitive interface on a smartphone that turns the phone into a turning movement counter. Because the phone provides constant connectivity to the internet, *TrafficTurk* allows for real–time data collection and analysis. This work is intended to eventually support such a system.

B. Related work

The problem of estimating properties of the traffic signal from sensor data has been examined by several authors. For example, [5] developed a method to estimate queue lengths at signalized intersections using travel times through the intersection, collected from mobile GPS sensors. Hofleitner et al. presented an unsupervised classification algorithm to infer the existence of a traffic signal on a road segment using sparse probe vehicle data [10]. Another related project is *SignalGuru* [14], which is a *Green Light Optimal Speed Advisory* (GLOSA) system that uses the camera and communication capabilities of windshield mounted smartphones to advise drivers about the optimal driving speeds in order avoid stopping at intersections. The system builds a database of fixed time signals, while adaptive signals are predicted with a support vector machine using a week long log of the adaptive signals. Our work differs from *SignalGuru*, both in the sensing (*TrafficTurk* measures vehicle maneuvers, from which the signal phase timing must be inferred [9], and not directly the traffic signal), and in the control objective estimation approach. The *SMART-SIGNAL* [15] system is another initiative that aims to collect high–resolution data from signalized intersections and use it to infer useful knowledge of the traffic system. This system communicates valuable traffic information which is often only available at the roadside signal controller, and may significantly improve the information available to traffic estimation systems in the future.

This work formalizes the problem of estimating the control strategies as an *inverse optimal control* (IOC) problem. Unlike optimal control, which computes a control policy that maximizes some performance objective under constraints [4], the inverse optimal control problem aims to estimate the unknown objective function given realizations of the system trajectory, which are assumed to be optimal. This is a useful concept in our problem, since we can use it to find an objective function under which the true system policy is optimal, thus recovering an optimal control program whose solution mimics the true system evolution. The inverse optimal control problem was studied as early as [12], and recently in the machine learning community as a related

[1]S. Gowrishankar is an M.S. Student in the Department of Mechanical Science and Engineering, University of Illinois at Urbana–Champaign, Urbana IL, 61801, USA. sgowris2@illinois.edu

[2]D. Work is an Assistant Professor in the Department of Civil and Environmental Engineering and the Coordinated Science Laboratory, University of Illinois at Urbana–Champaign, Urbana IL, 61801, USA. dbwork@illinois.edu

problem of *inverse reinforcement learning* (IRL). For example, Ng and Russell [16], Abbeel and Ng [2] have proposed methods that span applications including learning the control strategy for helicopter acrobatics and bipedal robots. In the realm of transportation systems, inverse reinforcement learning has been studied in relation to autonomous parking lot navigation [1], navigation and driving behaviors [21], helicopter flight [7], and hybrid vehicle fuel efficiency [20] among others.

However, as [3] points out, many approaches to inverse optimal control include the repeated solving of an optimal control problem within the inverse reinforcement learning framework and tend to be computationally intensive. Keshavarz et al. [13] propose a method to significantly reduce the computational requirements by posing the inverse optimal control problem as a convex optimization problem. Aghasadeghi et al. [3] extend the idea in [13] to solve an inverse optimal control problem for a hybrid system with impacts, which also inspired the approach taken in this paper.

In particular, optimal control approaches to traffic have been widely studied and part of our work is derived from the work of De Schutter [17] who models a single traffic intersection and solves the optimal control problem associated with it. We also use the idea proposed by Keshavarz et al. in [13] when solving the inverse optimal control problem for a traffic controller due to its computational efficiency.

C. Outline and contributions

The main contribution of this paper is the development of a computationally efficient method to solve the inverse optimal control problem for a traffic controller at an intersection via convex programming. More specifically, our method only requires solving a single convex optimization problem, instead of repeatedly solving several optimal control problems, common in many other methods. We first review the relevant components of the single intersection optimal control problem in Section II. In Section III, we build the objective function in terms of a basis of features with unknown weights and derive necessary and sufficient conditions for optimality of the optimal control problem. We then use the optimality conditions as constraints in the setup of the inverse optimal control problem. Finally, in Section IV, numerical experiments are performed on a single intersection to test the performance of the proposed method.

II. Intersection traffic model and optimal signal control

We review a model of traffic flow at a single intersection, and describe an optimal control problem to compute switching times of the traffic signal controller, originally proposed in [17]. The model and the optimal control problem are essential elements needed to build the inverse optimal control problem in Section III. A detailed analysis of the model and optimal control extensions can be found in [17]; we repeat only the relevant details here for completeness.

A. Continuous time dynamics of traffic flow at an intersection

Consider a model of a single intersection with m links indexed by i, managed by a traffic signal controller. The queue lengths $q_i(t)$ evolve in continuous time t, and k denotes the number of phase switches observed since the initial time t_0. The queue lengths $q_i(t)$ on each incoming link i at time t evolve according the following first order linear hybrid system:

$$\frac{dq_i(t)}{dt} = \begin{cases} a_{i,k} - \mu_{i,k} & \text{if } q_i(t) \geq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $a_{i,k} \geq 0$ denotes the arrival rate on link i during phase k, and $\mu_{i,k} \geq 0$ denotes the saturation rate on link i during phase k. We assume the saturation rate is zero when the light is red for link i during phase k.

In the model (1), the queue length function $q_i(t)$ is piecewise affine within a phase k. If t_k denotes the switching time when the k^{th} phase ends, the lengths of the queues at the switching times can be computed as

$$q_i(t_{k+1}) = \max\left\{\left((a_{i,k+1} - \mu_{i,k+1})\delta_{k+1} + q_i(t_k)\right), 0\right\}, \tag{2}$$

where $\delta_k = t_k - t_{k-1}$ be the duration of the k^{th} phase. To simplify our notation moving forward, let $\alpha_{i,k} = a_{i,k} - \mu_{i,k}$ denote the net flow into link i during phase k. The variable $q_{i,k} = q_i(t_k)$ denotes the queue length at time t_k on link i, and $q_k = [q_{1,k}, \cdots, q_{m,k}]^T$ is the vector of queue lengths at time t_k.

B. Optimal signal control via switching time control

With the model of the intersection defined, the finite horizon optimal control problem of minimizing some objective function $\tilde{J}$ over the $m \times (n+1)$ state variables $q = \left[{q_0}^T, \cdots, q_n^T\right]^T$ and the n control variables $\delta = [\delta_1, \cdots, \delta_n]^T$ can be written as the following *extended linear complementarity problem* (ELCP):

$$\begin{aligned}
&\text{minimize}_{q,\delta}: && \\
&\qquad \tilde{J}(q,\delta) && \\
&\text{subject to:} && \\
&q_{i,k+1} \geq \alpha_{i,k+1}\delta_{k+1} + q_{i,k} && \forall i \in \{1,\cdots,m\}, \\
&\qquad && \quad \forall k \in \{0,\cdots,n-1\} \\
&q_{i,k} \geq 0 && \forall i \in \{1,\cdots,m\}, \\
&\qquad && \quad \forall k \in \{1,\cdots,n\} \\
&(\alpha_{i,k+1}\delta_{k+1} + q_{i,k} - q_{i,k+1}) \times && \\
&\qquad (q_{i,k+1}) = 0 && \forall i \in \{1,\cdots,m\}, \\
&\qquad && \quad \forall k \in \{0,\cdots,n-1\} \\
&q_{i,0} = q_{i_\bullet} && \forall i \in \{1,\cdots,m\} \\
&\delta_k \geq \delta_{min} && \forall k \in \{1,\cdots,n\} \\
&\delta_k \leq \delta_{max} && \forall k \in \{1,\cdots,n\} \\
&\textstyle\sum_{k=1}^{n} \delta_k = t_f - t_0. &&
\end{aligned} \tag{3}$$

where $t \in [t_0, t_f]$ is given in terms of the initial time t_0 and the final time t_f. The parameters $\delta_{\min}$ and $\delta_{\max}$ are the

upper and lower bounds of the phase duration, and prevent the signal from switching too quickly or not at all. Note that (3) is nonlinear due to the following constraint:

$$\begin{aligned}(q_{i,k+1} - \alpha_{i,k+1}\delta_{k+1} - q_{i,k})\, q_{i,k+1} = 0,\\ \forall i \in \{1, \cdots, m\},\ \forall k \in \{0, \cdots, n-1\}\end{aligned} \tag{4}$$

which requires that either $(q_{i,k+1} - \alpha_{i,k+1}\delta_{k+1} + q_{i,k})$ or $q_i(t_{k+1})$ is equal to zero.

As identified in [17], problem (3) can be relaxed to a linear constraint set by dropping the complementarity constraints, yielding:

$$\text{minimize}_{q,\delta}: \quad \tilde{J}(q, \delta)$$

subject to:

$$\begin{array}{ll} q_{i,k+1} \geq \alpha_{i,k+1}\delta_{k+1} + q_{i,k} & \forall i \in \{1, \cdots, m\}, \\ & \quad \forall k \in \{0, \cdots, n-1\} \\ q_{i,k} \geq 0 & \forall i \in \{1, \cdots, m\}, \\ & \quad \forall k \in \{1, \cdots, n\} \\ q_{i,0} = q_{i\bullet} & \forall i \in \{1, \cdots, m\} \\ \delta_k \geq \delta_{min} & \forall k \in \{1, \cdots, n\} \\ \delta_k \leq \delta_{max} & \forall k \in \{1, \cdots, n\} \\ \sum_{k=1}^{n} \delta_k = t_f - t_0 & \end{array} \tag{5}$$

Note that in (5), the constraint set is linear, and therefore it can be written as:

$$\text{minimize}_{x \in \mathcal{X}}: \quad J(x)$$

where

$$x = \begin{bmatrix} q \\ \delta \end{bmatrix},$$

and

$$\mathcal{X} = \left\{ \tilde{x} \in \mathbb{R}^{(n+1)m+n} \mid A\tilde{x} \leq b, A_{\text{eq}}\tilde{x} = b_{\text{eq}} \right\},$$

for suitable A, A_{eq}, b, and b_{eq}.

An important result from [17] links problem (5) and (3). Specifically, if $\tilde{J}(q, \delta)$ is a strictly increasing function of the queue lengths q, [17] showed the optimal solution of (5) is also optimal for (3). Thus, for a restricted class of objective functions, the ELCP optimal control problem (3) can be solved with a tight convex relaxation.

C. Objective functions

Many convex objective functions can be considered for (3) and (5). For example, the sum of the squared queue length at switching times can be can be written as

$$J_1(x) = \tilde{J}_1(q) = \sum_{i=1}^{m} \sum_{k=0}^{n} q_{i,k}^2, \tag{6}$$

Because (6) is strictly increasing in q, the solution to (5) with objective (6) is also optimal for (3).

Noting the average phase duration over a finite interval $\bar{\delta} = \frac{t_f - t_0}{n}$ is a constant, the variance of the phase duration can be written as:

$$\text{var}(\delta) = \frac{1}{n-1} \sum_{k=1}^{n} \left(\delta_k - \bar{\delta}\right)^2. \tag{7}$$

Expanding (7) yields:

$$\text{var}(\delta) = \frac{1}{n-1} \left(\sum_{k=1}^{n} \delta_k^2 - n\bar{\delta}^2 \right),$$

which has the same minimizer as

$$J_2(x) = \tilde{J}_2(\delta) = \sum_{k=1}^{n} \delta_k^2. \tag{8}$$

Since (8) is not a strictly increasing function of q, the solution to (5) with objective (8) (equivalently (7)) will not in general be optimal for (3).

For the purpose of inverse optimal control, we can combine (6) and (8) for multi-objective optimization as

$$J(x) = \textstyle\sum_{j=1}^{2} w_j J_j(x), \tag{9}$$

where $w = [w_1, w_2]^T \in \mathcal{W}$, where $\mathcal{W} = \left\{ \tilde{w} \in \mathbb{R}^2 \,\middle|\, \sum_j \tilde{w}_j = 1, \tilde{w}_1 > 0 \right\}$, are the normalized weights. Since $w_1 > 0$, the solution to (5) with objective (9) is also optimal for (3).

Although many other features could be considered in the optimal control problem, we illustrate our examples on these particular features for two reasons. The first reason is that the features describe plausible objectives that a traffic controller would optimize over. The first feature (6) minimizes the queue lengths observed at the intersection at the switching times. It would be reasonable to assume that optimization over such an objective is feasible and useful for a traffic controller. The second feature minimizes the variance of the phase lengths and is useful to describe a traffic controller whose policy is to create phases of the same length. Because the features are convex, the corresponding optimal control problem can be solved efficiently.

III. Recovering switching control objectives via inverse optimal control

A. Inverse optimal control problem definition

The inverse optimal traffic signal problem can be stated as follows. Given an observation of the system trajectory $x^* = \begin{bmatrix} q^* \\ \delta^* \end{bmatrix}$, find the weights $w^* \in \mathcal{W}$ such that

$$x^* = \arg\min_{x \in \mathcal{X}} \textstyle\sum_{j=1}^{2} w_j^* J_j(x). \tag{10}$$

In other words, the system trajectory should be optimal for (10) under some weights w, which are to be estimated. Since the system trajectory x^* is assumed to be optimal, it must satisfy the *Karush–Kuhn–Tucker* (KKT) conditions for optimality. The conditions are given by:

$$\begin{aligned} \textstyle\sum_{j=1}^{2} w_j \nabla J_j(x) + A^T\lambda + A_{\text{eq}}^T\nu &= 0 \\ A_{\text{eq}}x - b_{\text{eq}} &= 0 \\ Ax - b &\leq 0 \\ \lambda^T(Ax-b) &= 0 \\ \lambda &\geq 0, \end{aligned} \tag{11}$$

where λ and ν are the Lagrange multipliers associated with the inequality and equality constraints respectively.

If $\delta_{\min} < \delta_{\max}$, the KKT conditions (11) are also sufficient conditions for (10) because *Slater's constraint qualification* holds [6]. This follows from the fact that an interior point $\hat{x} = \begin{bmatrix} \hat{q} \\ \hat{\delta} \end{bmatrix}$ can be easily constructed by selecting some $\delta_{\min} < \hat{\delta}_k < \delta_{\max}$ for all k, such that the phase durations $\hat{\delta}$ are strictly feasible. Then, for each link i, initialize the queue lengths according to the initial data $\hat{q}_{i,0} = q_{i\bullet}$. Now, the strictly feasible queue lengths can be computed according to $\hat{q}_{i,k+1} = \max\left\{\alpha_{i,k+1}\hat{\delta}_{k+1} + \hat{q}_{i,k}, 0\right\} + \varepsilon$, for some $\varepsilon > 0$. Because there are no upper bounds on the queue lengths, the evolution of the queues given by $\hat{q}$ is strictly feasible when computed in this way.

B. Convex inverse optimal control problem

Given the necessary and sufficient conditions for optimality, we can now use them to solve the inverse optimal control problem. Following the generalized inverse optimal control approach proposed in [13], we treat the KKT conditions for optimality as constraints to the inverse optimal control problem. This approach guarantees that the weights returned by the following optimal control problem are indeed the weights that satisfy the optimal control problem. Moreover, the optimization is itself a convex program, which can be solved without repeatedly solving the optimal control problem (10), common to many approaches.

The *approximately optimal* [13] inverse optimal control program is:

$$\text{minimize}_{r,\lambda,\nu,w}:$$

$$\textstyle\sum_{l=1}^{2} \|r_l\|_2^2$$

subject to :

$$\begin{gathered} \textstyle\sum_{j=1}^{2} w_j \nabla J_j(x^*) + A^T\lambda + A_{\text{eq}}^T\nu = r_1 \\ \lambda^T(Ax^* - b) = r_2 \\ \lambda \geq 0 \\ w \geq 0 \\ \textstyle\sum_{j=1}^{2} w_j = 1 \\ w_1 \geq \vartheta \end{gathered} \tag{12}$$

In (12) above, the decision variables are the weights w, the Lagrange variables λ and ν, and the residuals r. The objective minimizes the 2–norm of the residuals, and takes the value zero when the KKT conditions are exactly satisfied. In general, the conditions need not be satisfied, especially if there are errors in the model used within the inverse optimal control problem (e.g. the objective basis functionals do not span the space of the true control objective, the incoming and outgoing flows have some error, etc.). Instead, the KKT conditions are allowed to be approximately satisfied, which yields an approximately optimal inverse optimal control problem [13]. Also note that the weight associated with the queue length is not allowed to be exactly equal to zero ($w_1 \geq \vartheta$) in order to maintain the strictly increasing property of the objective function with respect to the queue length, where $1 >> \vartheta > 0$ is a small nonnegative number.

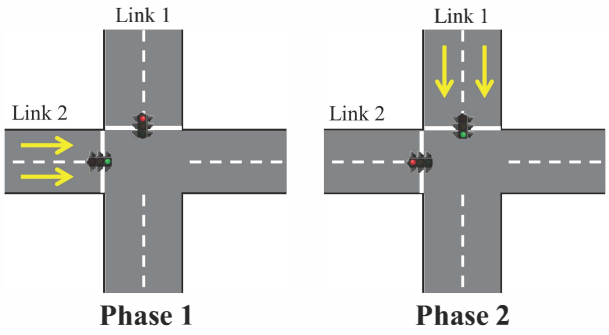

Fig. 1: Phases for two intersecting one–way streets

IV. Numerical Experiments

This section tests the performance of the inverse optimal control method to estimate the control objective (12) through numerical experiments. A single instance of the problem is examined, when the estimator has perfect knowledge of the true system dynamics. In practice, the arrival rates may have errors, therefore one instance of the problem with errors in the arrival rates is also tested.

A. Experimental setup

Consider an intersection of two one–way streets (Figure 1). The intersection has two traffic signal phases. The first phase lets vehicles on link 2 pass the intersection while stopping vehicles on link 1. Similarly, the second phase lets vehicles on link 1 pass the intersection while stopping vehicles on link 2. The saturation rates for both links are taken to be 1600 veh/hr/lane and since we consider two lanes on each link, we choose the total departure rate per link to be 3200 veh/hr when the light is green on that link. The true signal is observed over a period of 20 min (i.e. $t_0 = 0$, $t_f = 20$ min). The minimum phase length is taken to be $\delta_{\min} = 45$ seconds and the maximum phase length to be $\delta_{\max} = 3$ mins. The number of phases n is set at 15. Using the parameters in Table I, we generate synthetic (numerical) observations x^{true} from the true system by solving (10) under the weights w_j^{true}. The observations consist of the queue lengths at the switching times, as well as the switching times, so that the phase durations can be computed. Then, we solve the inverse optimal control problem to recover a set of estimated weights w_j^{est}. The estimated weights are then used to solve the optimal control problem once more in order to generate the an estimated trajectory x^{est}.

B. Error metrics

Several performance metrics are defined to assess the accuracy of the inverse optimal control solver. The *absolute percent error* on the true objective is computed in the following way:

$$e_{obj} = 100\times\frac{\left|\sum_{j=1}^{2} w_j^{true} J_j\left(x^{true}\right) - \sum_{j=1}^{2} w_j^{true} J_j\left(x^{est}\right)\right|}{\sum_{j=1}^{2} w_j^{true} J_j\left(x^{true}\right)}, \tag{13}$$

where w_1^{true} and w_2^{true} are the assumed weights of the features in the objective function of the true system, x^{true} is the true system trajectory resulting from solving the optimal control problem described in (10), and x^{est} is the estimator's trajectory. Therefore, $\sum_{j=1}^{2} w_j^{true} J_j\left(x^{true}\right)$ and $\sum_{j=1}^{2} w_j^{true} J_j\left(x^{est}\right)$ represent the objective values of the true system trajectory and the estimated system's trajectory with respect to the control objective of the true system. Therefore, this metric measures the difference in performance between the true system and estimator based on the control objective of the true system.

The second metric measures the error of the phase lengths under the estimated control objective compared to the phase lengths under the true control objective. The error in the phase lengths is computed as follows:

$$e_\delta = \|\delta^{true} - \delta^{est}\|_1 \tag{14}$$

where δ^{true} and δ^{est} are vectors of phase lengths of the true system and the phase lengths under the estimated control objective respectively.

Similarly, the errors in the queue lengths can be computed as:

$$e_q = \|q^{true} - q^{est}\|_1 \tag{15}$$

where q^{true} and q^{est} are vectors with the queue lengths generated by the true system and queue lengths under the estimated control objective.

C. Estimation with error–free model dynamics

In order to generate synthetic data, we solve the optimal control problem (5) with objective $\sum_{j=1}^{2} w_j J_j\left(x\right)$, with $w_1 = 0.9$, $w_2 = 0.1$ and J_1 and J_2 as defined above in (6) and (8). The arrival rates for all phases are $a_{1,k} = 1645$ veh/hr and $a_{2,k} = 123$ veh/hr on link 1 and 2 respectively and were drawn from the uniform distribution $\mathcal{U}[0, 2000]$. The experiment starts in phase 1 (Figure 1) and alternates between phases throughout the experiment.

The output is a sequence of phase lengths and queue lengths at different switching times. This observed trajectory is considered the true system trajectory which will be used for inversion. Since we model the objective function of the optimal control problem as a convex function, the optimal control solver converges quickly to the global optimum. The control policy generated by the optimal control solver is then provided to the estimator along with the model dynamics.

The model dynamics that are assumed by the estimator while solving the inverse optimal control problem are crucial in getting accurate results. We first test the case when the estimator has a model with exactly the same parameters as the true system. Since the synthetic data is generated from the same model of traffic (more specifically, the data comes from solving an optimal control problem with the same constraint set and objective function) that is used in the inversion, this numerical setting is referred to as an *inverse crime* [8], [11]. Regardless, it allows a best case performance of the method and can be used to assess the ill–posedness of the inverse optimal control problem.

Parameter	Link 1 and Link 2
m	2
n	15
$\delta_{\min}$	45 seconds
$\delta_{\max}$	3 min
$\mu_{i,k}$	$\begin{cases} 0 \text{ veh/hr} & \text{if red light} \\ 3200 \text{ veh/hr} & \text{if green light} \end{cases}$
t_f	20 min
$q_{i,\bullet}$	10 veh

TABLE I: Summary of true system parameters

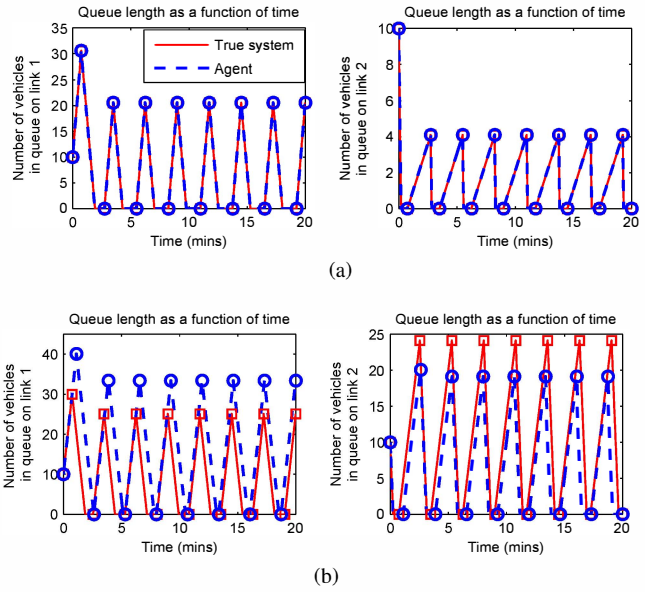

Fig. 2: Queue length evolution on link 1 (left) and link 2 (right) (a) true system (solid red) and estimator (dashed blue) when the true arrival rates are exactly known to the estimator (b) true system (solid red) and estimator (dashed blue) when the true arrival rates contain errors of 10%.

The IOC optimization problem (12) is solved, and the queue lengths (solid and dashed lines) and switching times (boxes and circles) of the true system and the estimator are shown in Figure 2a. The error in the value of the objective function recovered by the estimator is $e_{obj} = 0\%$. Figure 2a shows that the queue length evolution and switching times of the true system and estimator were almost identical when the model dynamics are known exactly to the estimator. The corresponding error metrics are $e_\delta = 0$ seconds, and $e_q = 0$ veh.

The inverse optimal control problem solves in approximately 25 seconds on when run in Matlab on a 64-bit

Windows 7 laptop with a dual-core 2.50GHz processor and 4.00GB of RAM. The optimal control problem is solved using *quadprog* and the inverse optimal control problem is solved with *fmincon* to simplify the coding requirements in our implementation, although any convex solver could be used.

D. Estimation with arrival rate errors

In practice, the arrival rate must be measured or estimated, for example by averaging the vehicle arrivals over a period of time. As a result, the arrival rates used in the inverse optimal control solver will contain errors compared to the rates in the true system. To synthetically create this situation, we introduce errors in the arrival rates that are provided to the estimator. Note that we assume the departure rate to remain the same and that we can accurately estimate this value.

In this experiment, we introduce a 10% error into the arrival rates assumed by the estimator and observe the trajectories produced by the true system and estimator. The parameters for the experiment are the same as in the first experiment with exception of the arrival rates which are again drawn from the uniform distribution $\mathcal{U}[0, 2000]$. The arrival rates for the true system were defined to be $a_{i,k}$ and now, suppose we define the arrival rate assumed by the estimator as $a^{est}_{i,k}$, then $a_{1,k} = 1600$ veh/hr, $a_{2,k} = 800$ veh/hr for the true system and $a^{est}_{1,k} = 1440$ veh/hr, $a^{est}_{2,k} = 720$ veh/hr for the estimator.

The queue lengths and switching times produced by the true system and estimator are shown in Figure 2b. The absolute percent error in the value of the objective function recovered by the estimator was $e_{obj} = 7.06\%$, the error in the queue lengths was $e_q = 102$ veh, and the error in the phase lengths was $e_\delta = 307$ seconds. It should be noted that even though the error in the weights is relatively large, the errors in queue lengths and phase lengths are smaller, given that approximately 1066 vehicles passed the intersection over the 20 min simulation.

The same experiment was repeated over a longer time horizon of 40 mins in order to test whether the performance of the estimator improves when more data is available to it. The error in the queue lengths and the error in the phase lengths decreased to $e_q = 91$ veh and $e_\delta = 208$ seconds. The error in the value of the objective function of the estimator also decreased to $e_{obj} = 6.04\%$.

V. Conclusions and future work

Traffic controllers are an integral part of any modern urban traffic system and understanding their control strategies is essential to predicting future states of traffic from observed data. However, it is generally challenging to procure the control strategies of traffic controllers in large scales when it is available, and in other cases it is impossible, due to proprietary algorithms or human–based traffic controllers. The work in this paper is a first attempt at recovering control strategies through inverse optimal control. The next steps to be taken include testing on intersections with more phases, and to create a rich set of basis functions for the objective function. The eventual goal is to test the performance of the inverse optimal control formulation experimentally on fixed–time and adaptive signals using data obtained from *TrafficTurk*. The possibility of coordination amongst traffic signals might also be explored.

References

[1] P. Abbeel, D. Dolgov, A. Y. Ng, and S. Thrun. Apprenticeship learning for motion planning with application to parking lot navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008*, pages 1083–1090. IEEE, 2008.

[2] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

[3] N. Aghasadeghi, A. Long, and T. Bretl. Inverse optimal control for a hybrid dynamical system with impacts. In *2012 IEEE International Conference on Robotics and Automation*, pages 4962–4967. IEEE, 2012.

[4] M. A. Athans and P. L. Falb. *Optimal Control: An Introduction to the Theory and Its Applications*. Dover Publications, 2007.

[5] X. Ban, P. Hao, and Z. Sun. Real time queue length estimation for signalized intersections using travel times from mobile sensors. *Transportation Research Part C: Emerging Technologies*, 19(6), 2011.

[6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[7] A. Coates, P. Abbeel, and A. Y. Ng. Apprenticeship learning for helicopter control. *Communications of the ACM*, 52(7):97–105, July 2009.

[8] D. Colton and R. Kress. *Inverse acoustic and electromagnetic scattering theory*, volume 93. Springer, 2013.

[9] M. R. Gahrooei and D. B. Work. Estimating traffic signal phases from turning movement counters. Submitted to 16th International IEEE Conference on Intelligent Transport Systems, April 2013.

[10] A. Hofleitner, E. Come, L. Oukhellou, J. Lebacque, and A. Bayen. Automatic inference of map attributes from mobile data. In *15th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1687–1692, September 2012.

[11] J. P. Kaipio and E. Somersalo. *Statistical and computational inverse problems*, volume 160. Springer Science+Business Media, 2005.

[12] R. E. Kalman. When is a linear control system optimal? *Journal of Fluids Engineering*, 86:51–60, 1964.

[13] A. Keshavarz, Y. Wang, and S. Boyd. Imputing a convex objective function. In *IEEE International Symposium on Intelligent Control (ISIC)*, pages 613–619. IEEE, 2011.

[14] E. Koukoumidis, L.-S. Peh, and M. Martonosi. Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 353–354. ACM, 2011.

[15] H. X. Liu, X. Wu, W. Ma, and H. Hu. Real-time queue length estimation for congested signalized intersections. *Transportation research part C: emerging technologies*, 17(4):412–427, 2009.

[16] A. Y. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the seventeenth international conference on machine learning*, pages 663–670, 2000.

[17] B. De Schutter. Optimal control of a class of linear hybrid systems with saturation. *SIAM Journal on Control and Optimization*, 39(3):835–851, 2000.

[18] T. Standage. *Mechanical Turk: The True Story of the Chess Playing Machine That Fooled the World*. Penguin Press, 2002.

[19] TrafficTurk. http://www.trafficturk.com, 2013.

[20] A. Vogel, D. Ramachandran, R. Gupta, and A. Raux. Improving hybrid vehicle fuel efficiency using inverse reinforcement learning. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 384–390, 2012.

[21] B. D. Ziebart, A. Maas, A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence, AAAI'08*, pages 1433–1438, 2008.