

Efficient Signal Control for Urban Traffic Networks with Unknown System Parameters

Negar Mehr¹, Jennie Lioris², Roberto Horowitz¹, and Ramtin Pedarsani³.

Abstract—Among the several signal control strategies that have been proposed in the literature, a key assumption is that system parameters including network service rates are known. However, it is envisaged that in the next generation of transportation networks with mixed autonomy, system parameters such as service rates vary as autonomy fraction changes. Aligned with this, we propose a signal control strategy in which, unlike previous approaches, both mean network demands and service rates of the queues are unknown to the controller. To this end, we use stochastic gradient projection to develop a cyclic iterative control, where at every cycle, the timing plan of the network signals is updated based on the measured changes in the network queue lengths such that the iterative signal control scheme is guaranteed to converge to an optimal signal plan, provided that the network demand and service rates are constants. We describe the intuition behind our algorithm, and further demonstrate through simulation studies, that our iterative control scheme can successfully stabilize the system.

I. INTRODUCTION

The rise of traffic congestion and demands in metropolitan areas has highlighted the importance of traffic management and control for every commuter. Traffic control strategies that increase throughput and mobility throughout the network result in reductions in delays, fuel consumption, and air pollution. Hence, they can lead to more sustainable infrastructures. In this work, we consider the design of signal control strategies for urban networks, which are the main factor in affecting urban traffic patterns. As a result of this importance, a large body of literature has been focusing on this problem, ranging from fixed time controllers to temporal logic based controls. Fixed time control refers to the class of periodic controls that operate on a cycle time such that each set of non-conflicting phases at an intersection, known as a stage, receives a fixed duration of green. A thorough analysis of fixed time control strategies is presented in [1].

Because of the simplicity of such controls and their practicality, several methods have been proposed for determining the green durations that each stage must receive during a cycle. Examples of such methods include SYNCHRO [2], VISGAOST [3], SCOOT [4], and OPAC [5]. SYNCHRO and VISGAOST utilize historical data to determine the signal timing plans in an offline fashion. SCOOT and OPAC decide

on the timing plan of each intersection such that a performance metric of upstream queues at each intersection is maximized. Another type of signal control that is shown to be effective in stabilizing traffic networks is Max Pressure [6]. In Max Pressure (MP), the stage with the highest pressure is actuated, where the pressure for each stage is computed using the length of the neighboring queues. Max pressure is proven to maximize the network throughput while guaranteeing stability. Although MP does not require the knowledge of the demands, a key assumption in its derivation is that service rates of the network queues are known.

A popular class of control strategies for signal control of urban networks and freeways is model predictive control (MPC) [7], [8], [9], [10]. Such controls are popular since they can systematically handle constraints. Using such controls, at the beginning of each cycle, the control scheme finds the timing plans such that a sensible performance metric of the system is optimized over a finite horizon, the obtained timing plan is implemented, and the same procedure is repeated. It is well known that in order for the MPC laws to work properly, accurate knowledge of the system model and parameters is required. However, obtaining such accurate knowledge of system parameters for transportation networks is nontrivial in general. Recently, with the advances in synthesis tools in formal methods, temporal logic tools have also been utilized for the task of traffic control [11], [12]. The viability of such methods depends on the existence of direct temporal specifications that describe the desired behavior of the system as well as accurate knowledge of system parameters.

The common feature of the majority of the aforementioned works is that they assume that network parameters such as service rates of the queues and demands are known to the controller. However, this might not necessarily be the case in practice. As an example, connected vehicle technology (CVT) which has recently gained a lot of attention and interest is going to be used for creating platoons of vehicles. It is shown that dependent on the penetration rate or the fraction of cars equipped with autonomy on the road, the service rate of the network queues changes [13], [14]. This implies that the higher the penetration rate, the higher the service rate will be. Hence, previous approaches which consider service rates to be known fixed parameters cannot be directly applied for such networks. Therefore, it is important to come up with strategies that are robust to such system parameters. In addition to the service rate, network demands might not be known in practice, as multiple algorithms for estimating the network demands have been developed [15], [16]. This further points to the superiority of approaches that are robust

¹N. Mehr and R. Horowitz are with the Mechanical Engineering Department, University of California, Berkeley, 94720 CA, USA. negar.mehr@berkeley.edu, horowitz@berkeley.edu

²J. Lioris is with the ENPC ParisTech, Paris, France jennie.lioris@cermics.enpc.fr

³R. Pedarsani is with the Department of Electrical and Computer Engineering at the University of California, Santa Barbara, Santa Barbara, CA 93106, USA. ramtin@ece.ucsb.edu

to the knowledge of network demands.

To address the need for taking the uncertainties in the parameters of transportation networks such as demands into account, in [17], a robust signal control is proposed. It considers the control in a daily basis, where the possible network demands are considered to form a discrete set of possible scenarios, and a robust optimization problem is solved to find the signal timing plans. In [18], several robust optimization problems are formulated to capture the tradeoff between optimality and stability when demands are subject to fluctuations. In [19], a model predictive control is designed where the effects of the unknown bounded demand on the optimal solution is considered.

In this paper, we propose a control strategy that is robust to the knowledge of both the network service rates and demands similar to [20], [21], [22]. In the absence of knowledge of the demands and service rates, we determine the green durations required at every cycle such that the control converges to the desired timing plan for maximizing the throughput of the network. Our approach is different from the previous work in that it *learns* the timing plan iteratively by measuring the changes in queue lengths rather than bounding the uncertainty and planning for the worst case scenario or the average case. In our approach, we assume that turning ratios of the network are known, but this is not a restrictive assumption since such information is usually obtained via surveys or counts [23]. We use the PointQ model [6] to formally define the urban network as a queuing system. We state the requirements that our synthesized control must satisfy in order to be implementable as a cyclic control. We describe how the green times that different movements receive during each cycle can be updated using the changes in queue lengths with a gradient projection algorithm such that all the flows in the network are balanced. Such approach guarantees that the iterative control scheme converges to an optimal and “balanced” signal plan, provided that the network parameters are constants. We further demonstrate the capability and performance of our algorithm in our simulations.

The organization of this paper is as follows. In Section II, we describe the notation we use. In Section III, we present the network model. In Section IV, we provide the description of our iterative robust control algorithm. We demonstrate the practicality and performance of our method in simulation examples in Section V. We conclude the paper and discuss possible future directions in Section VI.

II. NOTATION

For a vector $x \in \mathbb{R}^n$, we use x_i to represent its i_{th} element. For any vector, inequalities are interpreted element-wise unless otherwise mentioned. We let $\mathbb{R}_+^n = \{x \in \mathbb{R}^n, 0 \leq x_i, \forall 1 \leq i \leq n\}$ to be the set of real n dimensional vectors with positive elements. The inequalities involving vector are interpreted as elementwise inequalities. To distinguish matrices from vectors, we show matrices with upper case letters such as X . For a matrix X , X_{ij} is the element on the i_{th} row and j_{th} column of X . We let X^T denote matrix

transpose. For a set \mathcal{S} , $|\mathcal{S}|$ is the cardinality of the set \mathcal{S} . Moreover, for any vector x , $[x]_{\mathcal{S}}$ is the convex projection of x on the set \mathcal{S} .

III. NETWORK MODEL

In order to model system evolution and dynamics, we use PointQ model [6]. PointQ models traffic networks as store-and-forward queuing systems. It allows for characterizing feasible demand profiles and stabilizing signal controls. Consider the network graph to be $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where \mathcal{N} is the set of graph nodes, and \mathcal{L} is the set of graph edges. The set of nodes \mathcal{N} and edges \mathcal{L} represent network intersections and links respectively. Let $|\mathcal{N}| = N$ be the number of network nodes, and $|\mathcal{L}| = L$ be the number of network links.

In a traffic network, links are divided into three types: entry links $\mathcal{L}_{\text{entry}}$, internal links $\mathcal{L}_{\text{inter}}$, and exit links $\mathcal{L}_{\text{exit}}$. Entry links are the links that carry exogenous arrivals to the network. They are identified by the fact that entry links do not have any starting nodes in the network. Internal links connect network nodes; hence, they have both starting and end nodes. Finally, exit links are the ones through which vehicles leave the network and do not have any end nodes in the network. We assume that network demand and saturation flow rates are random. For each link in the network $l \in \mathcal{L}$, let f_l represent the long run average of the flow of the vehicles that leave link l , and d_l be the mean value of the exogenous arrivals on link l respectively. Note that since we have assumed that exogenous arrivals enter the network only through entry links, $d_l = 0, \forall l \in \mathcal{L}_{\text{inter}} \cup \mathcal{L}_{\text{exit}}$.

For each node in the network, only certain movements or phases are allowed. For each of such movements, a separate queue is considered in PointQ. In fact, we use queues and movements interchangeably in this paper. Each movement is characterized by its origin and destination links. Moreover, for each pair of links $l, m \in \mathcal{L}$, $r(l, m)$ is the fraction of vehicles joining link m when leaving link l , or equivalently, the probability that random vehicle joins link m from link l . It is assumed that $r(l, m)$'s are known a-priori. We use $\mu(l, m)$ to represent the mean saturation flow rate or service rate of the movement from link l to m .

At each node $n \in \mathcal{N}$, let $I(n)$ and $O(n)$ be the incoming and outgoing links of node n . Since at each network node, flow conservation holds, we have

$$\sum_{l \in I(n)} f_l = \sum_{m \in O(n)} f_m, \quad \forall n \in \mathcal{N}. \quad (1)$$

Furthermore, for each l, m , and $o \in \mathcal{L}$, the link flows and the movement flows must satisfy the following:

$$\begin{aligned} f_l &= d_l, & f(l, m) &= r(l, m)f_l, & \text{if } l \in \mathcal{L}_{\text{entry}}, & (2) \\ f_l &= \sum_{o \in \mathcal{L}} f(o, l), & f(o, l) &= r(o, l)f_o, & \text{if } l \in \mathcal{L}_{\text{inter}} \cup \mathcal{L}_{\text{exit}}. & (3) \end{aligned}$$

With this background in mind, we proceed to how we synthesize the cyclic controls.

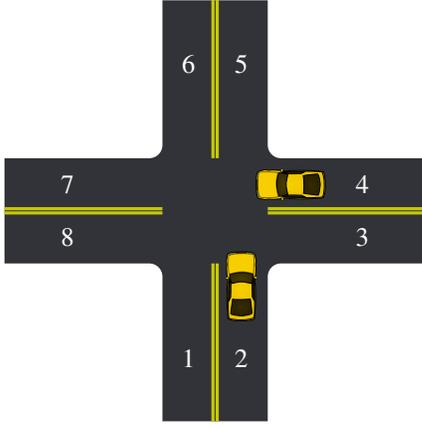


Fig. 1. Schematic of an intersection.

A. Cyclic Control

Assume that all actuators of the network of signalized intersections are cyclic controllers that operate on a common cycle time T . For such controllers, for each node $n \in \mathcal{N}$, there exist multiple stages $s_j^n, 1 \leq j \leq S^n$, where S^n is the total number of stages at node n . Each stage s_j^n is a set of non-conflicting movements that can be actuated simultaneously. Assume that stage s_j^n receives green for g_j^n fraction of the cycle time. Then, for a movement from link l to m belonging to the stage s_j^n , let $g_j^n(l, m)$ be the fraction of the green time that this movement receives during stage s_j^n . Thus, if multiple movements are actuated during a stage s_j^n , the green durations that they receive during stage s_j^n must be equal. In other words, if movements from links l and u to links m and v are two of such movements that belong to s_j^n , we have

$$g_j^n(l, m) = g_j^n(u, v) = g_j^n, \quad \forall (l, m) \text{ and } (u, v) \in s_j^n. \quad (4)$$

Note that all stage green fractions g_j^n must add up to 1 at each node n , so we have

$$\sum_{j=1}^{S^n} g_j^n = 1, \quad \forall n \in \mathcal{N}. \quad (5)$$

Remark 1. In Equation (5), no clearance time between stages is considered. If clearance times must also be taken into account, g_j^n 's add to $1 - \epsilon^n$, where ϵ^n is the fraction of the cycle time during which "all red" undergoes at node n .

It might be the case that a movement receives green during multiple stages. For each movement from l to m , we let $p(l, m)$ to be the fraction of the aggregate green durations that this movement receives during each cycle:

$$p(l, m) = \sum_{j=1}^{S^n} g_j^n(l, m). \quad (6)$$

Example: To illustrate the notation, consider the schematic intersection in Figure 1. There is only one node. Let this node be indexed by 1. Thus, superscript 1 is considered. The intersection has 8 links with links 2, 4, 6, and 8 being

entry links, and 1, 3, 5 and 7 being exit links. There is no internal link in this example. If there exist only through and right movements, the origin-destination links for all network queues are as follows: (2,5), (4,7), (8,3), (6,1), (2,3), (4,5), (6,7) and (8,1). Assume that there are only 2 stages at the intersection such that each of which lasts half of the cycle time. The movements which are actuated during each stage are the following:

First Stage: (2,5), (4,5), (2,3), (6,1), (8,1), and (6,7).

Second Stage: (4,7), (6,7), (4,5), (8,3), (8,1), and (2,3).

Then, using our notation, for the first stage, we have

$$g_1^1(2, 5) = g_1^1(4, 5) = g_1^1(2, 3) = g_1^1(6, 1) = g_1^1(8, 1) = g_1^1(6, 7) = 0.5,$$

while, for the second stage, we have

$$g_2^1(4, 7) = g_2^1(6, 7) = g_2^1(4, 5) = g_2^1(8, 3) = g_2^1(8, 1) = g_2^1(2, 3) = 0.5.$$

We can encode the requirement in (5) as follows:

$$g_1^1(2, 5) + g_2^1(4, 7) = 1.$$

Finally, the aggregate green ratios of the movements are defined as:

$$p(2, 5) = p(6, 1) = p(4, 7) = p(8, 3) = 0.5, \\ p(4, 5) = p(2, 3) = p(8, 1) = p(6, 7) = 1.$$

B. Compact Notation of the Model

To increase the readability of the paper when describing our signal control algorithm, we introduce a compact notation of the introduced model quantities. We use similar notation to the one in [24]. We use $d \in \mathbb{R}_+^L$ and $f \in \mathbb{R}_+^L$ to represent the vectors of mean demands and average flows for all links in the network. Each element d_l of the vector d is simply equal to the mean exogenous arrival on link l if l is an entry link and zero otherwise. We can also collect the turning probabilities $r(l, m)$'s into the matrix $R \in \mathbb{R}^{L \times L}$ such that $R_{lm} = r(l, m)$. Using this notation, the set of Equations (2) and (3) can be simply written as:

$$f = (I - R^T)^{-1}d. \quad (7)$$

In addition to the vector of link flows, we can construct the vector of average movement flows. Assume that there exists a total of B possible movements in the network. We let $\varphi \in \mathbb{R}_+^B$ be the vector of movement flows $f(l, m), \forall l, m \in \mathcal{L}$ for all allowed movements in the network. Moreover, we collect the aggregate fraction of green that each movement receives for all network movements, $p(l, m)$'s, in the allocation vector $p \in \mathbb{R}^B$. We further collect the service rates or mean saturation flows of all network queues, $\mu(l, m)$'s, in the diagonal matrix M whose i th diagonal element is equal to the service rate of the i th movement.

Using Equations (2) and (3), it is easy to see that the linear mapping from the vector of average link flows f to the vector of average movement flows φ can be encoded using a matrix $\Gamma \in \mathbb{R}^{B \times L}$:

$$\varphi = \Gamma f, \quad (8)$$

where at each row of Γ , say j_{th} row, all elements are zero except for the j_{th} element which is equal to $r(l, m)$ with l and m being the origin and destination links of the j_{th} movement. A given demand pattern is called feasible if and only if the mean demand vector d is such that for every $l, m \in \mathcal{L}$, $f(l, m) < \mu(l, m)p(l, m)$. Using our vectorized notation a demand vector is feasible if

$$\Gamma(I - R^T)^{-1}\lambda \leq Mp. \quad (9)$$

Next, we aim to represent the conditions in Equations 4, (5), and (6) via linear vector equalities. To this end, we collect all fraction of greens that all movements receive during the signal stages, $g_j^n(l, m)$'s in the vector $g \in \mathbb{R}^H$, where H is the total number of movement stage green fractions for all movements and all intersections. Then, we can rewrite Equation (6) for all queues as:

$$p = A_{g \rightarrow p}g, \quad (10)$$

where $A_{g \rightarrow p}$ is the linear mapping of appropriate dimension for maintaining (6). Moreover, we define the linear mapping A_{eq} to encode (4) for all pairs of queues that belong to the same stage at a single intersection by:

$$A_{\text{eq}}g = \mathbf{0}^{N \times 1}. \quad (11)$$

Finally, we use the linear transform A_{sum} in order to enforce (5) for all the nodes in the network:

$$A_{\text{sum}}g = \mathbf{1}^{N \times 1}. \quad (12)$$

Therefore, the requirements imposed by the cyclic implementation of the signal control are represented by Equations (10), (11), and (12).

The last component of the vectorized notation of our model quantities and parameters, is the vector of the queue lengths for all network queues. In the remainder of the paper, we let $q \in \mathbb{R}^B$ to denote the vector of queue lengths for all queues in the network.

IV. ROBUST NETWORK CONTROL

Before we proceed to the description of our algorithm, we need to describe some notations and definitions we need. We define the convex set \mathcal{C} to be the following:

$$\mathcal{C} = \{p \in \mathbb{R}^B \mid \exists g \geq 0, \text{ such that } p = A_{g \rightarrow p}g, \quad (13)$$

$$A_{\text{eq}}g = 0, A_{\text{sum}}g = \mathbf{1}\}.$$

The set \mathcal{C} is indeed the set of all p 's for which there exists a corresponding vector g that satisfies the constraints required by cyclic implementation of the signal control. We let $E \in \mathbb{R}^{B \times B}$ be a diagonal matrix with each diagonal entry e_{ii} being 1 if the i_{th} queue is nonempty and zero otherwise. Furthermore, we let k represent the time step index. Every time step of the control is assumed to last a cycle time T . Therefore, the time at which the k_{th} cycle begins is equal to kT . Now that we have introduced k , we let $q(k)$ to be the vector of queue lengths for all network queues at the end of the cycle k .

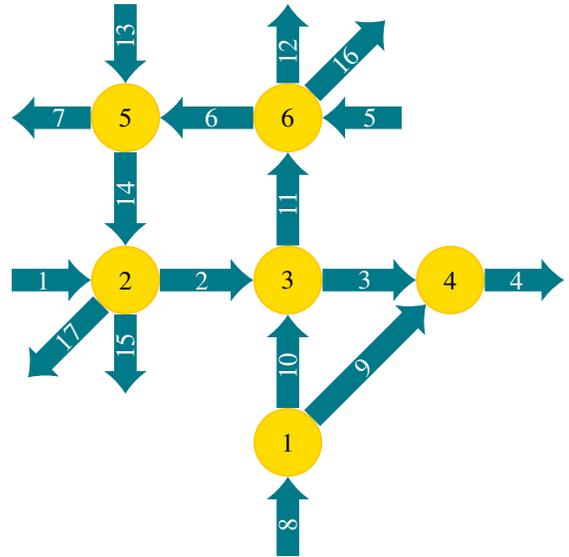


Fig. 2. The topology of the example network.

Using the cycle index k , we also define the sequence of step sizes or learning rates, $\{\beta(k)\}$, to be a decreasing sequence such that

$$\begin{aligned} \beta(k) &\rightarrow 0 \text{ as } k \rightarrow \infty, \\ \sum_{k=0}^{\infty} \beta(k) &= \infty, \\ \sum_{k=0}^{\infty} \beta(k)^2 &< \infty. \end{aligned} \quad (14)$$

The above conditions on $\beta(k)$ are standard assumption required for convergence of the stochastic gradient projection algorithm. As an example of such sequences, $\frac{1}{k}$ satisfies the above conditions. We also use $\Delta q(k) \in \mathbb{R}^B = q(k) - q(k-1)$ to denote the vector of the difference in queue length at the end and beginning of the k_{th} cycle, where $k \geq 1$. At last, we define the matrix $\Lambda \in \mathbb{R}^{L \times B}$ to be such that at the i_{th} row of the matrix, all elements are zero except for the elements located at the j_{th} columns with j being the index of the queues that originated from the link i . For such elements, $\Lambda_{ij} = 1$.

Example: Consider a network for which there are two queues from link 2 to links 5, 8. Assume that the indexes of queues for the movements (2, 5) and (2, 8) are 3 and 6 respectively. Then, the elements Λ_{23} and Λ_{26} are equal to 1, while other elements of the second row of Λ are zero. Each row of Λ can be constructed likewise.

With the introduced notation, we are ready to state our iterative control algorithm. At the beginning of every cycle k , we update p as follows:

- 1) Initialize $p(0)$ with an arbitrary feasible value (such that $p(0) \in \mathcal{C}$).
- 2) At each time step $k, k \geq 1$, update the vector p via the

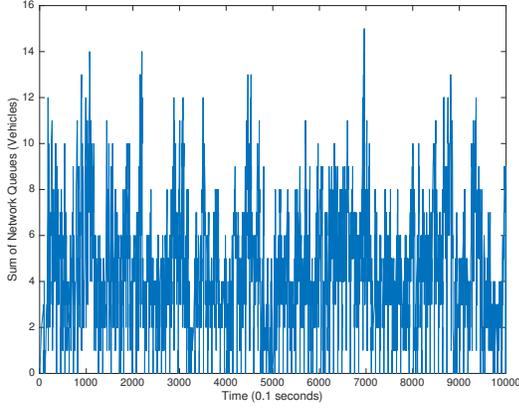


Fig. 3. Sum of all queues vs. time.

following:

$$p(k) = [p(k-1) + \beta(k)\Gamma E(I - R^T)^{-1}\Lambda\Delta q]_{\mathcal{C}}, \quad (15)$$

where $[\cdot]_{\mathcal{C}}$ is the projection on the set \mathcal{C} .

- 3) Apply the updated control $p(k)$ to the system, and let the system evolve to the next cycle time; then, measure $\Delta q(k)$ and repeat step 2.

It is important to mention that in step 3, for implementing a control law, we need $g(k)$ rather than $p(k)$. In fact, once $p(k)$ is found, the vector $g(k)$ for which $p(k) = A_{g \rightarrow p}g(k)$ is needed for implementing the new timing plan. Since there might be multiple $g(k)$ such that $p(k) = A_{g \rightarrow p}g(k)$, we can obtain $g(k)$ by simply solving a least squares problem

$$g(k) = \min_x \|p - A_{g \rightarrow p}x\|^2.$$

Note that the implicit assumption in the proposed algorithm is that the vector of the network mean demands d remains constant although its value is unknown. However, even if the mean value of the network demands changes, as long as the changes in d are slow enough, the algorithm can learn to adapt itself to the new demand profile. Moreover, we assume that the set of network turn ratios R are known. As we stated, this is not a restrictive assumption as such measures might be easily obtained through surveys or other simple estimation methods [25].

We next explain the intuition behind our algorithm. If the vector of the mean demand values d and, hence, link flow f , and the service rates M were known, then, the gradient projection update rules of the form

$$p(k+1) = [p(k) - \beta(k)(\Gamma f - Mp(k))]_{\mathcal{C}}$$

would have solved the following optimization problem:

$$\begin{aligned} & \underset{p}{\text{minimize}} && \frac{1}{2}\|\varphi - Mp\|^2 \\ & \text{subject to} && p \in \mathcal{C}. \end{aligned} \quad (16)$$

Note that the solution of the optimization problem yields the optimal signal plan that balances all the flows in the network

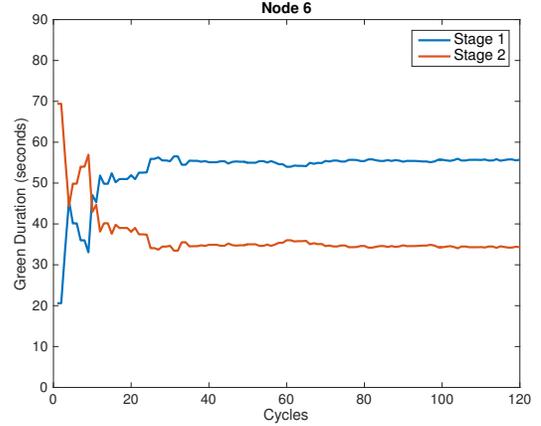


Fig. 4. Green duration of stages in node 6.

guaranteeing rate stability of the queues. However, in the absence of the knowledge M and λ for computing φ , we use $\Delta q(k)$ to estimate $(\Gamma f - Mp(k))$, as $\Delta q(k)$ is indeed an unbiased estimator of the gradient term with finite variance. Therefore, our update rule in (15) is a stochastic gradient projection algorithm for a convex optimization problem, which is guaranteed to converge given appropriate choice of the step size in (14). Moreover, as $p(k)$ converges to the optimal signal plan p^* that is the solution of (16), one can show that the network queues remain stable using similar proof as the one in [22].

It is noteworthy that, in general, queue lengths might not be available for all queues in the network. But, queue estimation methods have been proposed in literature using loop detectors, GPS data or combination of the two for queue estimation [26], [27]. Hence, when the queue lengths are not available, such methods can be utilized, and estimates of $\Delta q(k)$ can be considered for our algorithm. Clearly, the next step would be to evaluate the practicality of the above algorithm when such queue estimation methods are used.

V. SIMULATION RESULTS

To illustrate the performance of our control algorithm, consider the network shown in Figure 2. The network has 6 nodes, 17 links, and 20 queues. All network intersections have a common cycle time of 90 seconds. All nodes have cyclic controllers with known stages. Nodes 2,3,5 and 6 have 2 stages. The turning probabilities of the network queues are known to the controller. We assumed that the mean demand of the network and network service rates are unknown to the controller in the presence of random demands. For the simulation purpose, we used a typical set of mean demands for the network while they remained unknown to the controller.

We ran our control algorithm in closed loop with the simulation environment for 120 cycles. We started from a set of arbitrary yet feasible vector of stage and movement green durations. As Figure 3 demonstrates, the network queues remained stable (the sum of all queues in the network is not

growing and remains bounded). In addition to the network stability, the convergence of the network timing plans is demonstrated in Figure (4). Figure 4 shows such convergence of the stage green duration for node 6. Using Figure 4, we observe that the control learned a stabilizing timing plan over time despite the arbitrarily provided initial timing plan. Moreover, the convergence occurred quite fast, around the 40th cycle. The simulation was carried out for longer durations so as to showcase that convergence was achieved. Similar behavior and convergence results were achieved for other network nodes too. During our simulations, we found that convergence and stability were achieved for other feasible demands too that we omit reporting due to lack of space and similarity of behavior.

VI. CONCLUSION AND FUTURE WORK

In conclusion, we developed an iterative signal control for network of signalized intersections such that the control is unaware of the network mean demands and service rates of the queues. Our control is iterative in the sense that at the beginning of every cycle time, it decides on the next timing plan of the intersections based on the measured changes in network queue lengths using stochastic gradient projection. We showed the convergence of our algorithm and demonstrated through simulation studies that our controller can successfully stabilize the system.

The next step of this work will be to evaluate the performance of our algorithm when estimates of the queue lengths are used. Also, investigating the practicality of the control when the mean value of the demand changes rapidly over time will be important. In particular, finding the maximum tolerable rate of changes in the demand profiles would be of interest. Moreover, since the current approach is a centralized approach, it will be interesting to study how the designed control can be implemented using decentralized or distributed optimization techniques.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation under Grants CPS 1446145 and CPS 1545116 and the startup grant for Ramtin Pedarsani.

REFERENCES

- [1] A. Muralidharan, R. Pedarsani, and P. Varaiya, "Analysis of fixed-time control," *Transportation Research Part B: Methodological*, vol. 73, pp. 81–90, 2015.
- [2] D. Husch and J. Albeck, "Synchro 6: Traffic signal software, user guide," *Albany, California*, 2003.
- [3] J. Stevanovic, A. Stevanovic, P. T. Martin, and T. Bauer, "Stochastic optimization of traffic control and transit priority settings in vissim," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 3, pp. 332–349, 2008.
- [4] D. I. Robertson and R. D. Bretherton, "Optimizing networks of traffic signals in real time-the scoot method," *IEEE Transactions on vehicular technology*, vol. 40, no. 1, pp. 11–15, 1991.
- [5] N. H. Gartner, F. J. Pooran, and C. M. Andrews, "Implementation of the opac adaptive control strategy in a traffic signal network," in *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pp. 195–200, IEEE, 2001.
- [6] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.
- [7] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn, "Efficient network-wide model-based predictive control for urban traffic networks," *Transportation Research Part C: Emerging Technologies*, vol. 24, pp. 122–140, 2012.
- [8] S. Koehler, N. Mehr, R. Horowitz, and F. Borrelli, "Stable hybrid model predictive control for ramp metering," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pp. 1083–1088, IEEE, 2016.
- [9] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn, "Fast model predictive control for urban road networks via milp," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 846–856, 2011.
- [10] N. Mehr and R. Horowitz, "Probabilistic freeway ramp metering," *rN*, vol. 1, no. f1, p. 1, 2016.
- [11] S. Coogan, E. A. Gol, M. Arcak, and C. Belta, "Traffic network control from temporal logic specifications," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 2, pp. 162–172, 2016.
- [12] N. Mehr, D. Sadigh, R. Horowitz, S. S. Sastry, and S. A. Seshia, "Stochastic predictive freeway ramp metering from signal temporal logic specifications," in *American Control Conference (ACC), 2017*, pp. 4884–4889, IEEE, 2017.
- [13] J. Lioris, R. Pedarsani, F. Y. Tascikaraoglu, and P. Varaiya, "Platoons of connected vehicles can double throughput in urban roads," *Transportation Research Part C: Emerging Technologies*, vol. 77, pp. 292–305, 2017.
- [14] D. A. Lazar, S. Coogan, and R. Pedarsani, "Capacity modeling and routing for traffic networks with mixed autonomy," in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*, pp. 5678–5683, IEEE, 2017.
- [15] C.-J. Wu, T. Schreiter, and R. Horowitz, "Multiple-clustering armax-based predictor and its application to freeway traffic flow prediction," in *American Control Conference (ACC), 2014*, pp. 4397–4403, IEEE, 2014.
- [16] L. L. Ojeda, A. Y. Kibangou, and C. C. De Wit, "Adaptive kalman filtering for multi-step ahead traffic flow prediction," in *American Control Conference (ACC), 2013*, pp. 4724–4729, IEEE, 2013.
- [17] S. V. Ukkusuri, G. Ramadurai, and G. Patil, "A robust transportation signal control problem accounting for traffic dynamics," *Computers & Operations Research*, vol. 37, no. 5, pp. 869–879, 2010.
- [18] Y. Yin, "Robust optimal traffic signal timing," *Transportation Research Part B: Methodological*, vol. 42, no. 10, pp. 911–924, 2008.
- [19] T. Tettamanti, T. Luspai, B. Kulcsar, T. Peni, and I. Varga, "Robust control for urban road traffic networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 385–398, 2014.
- [20] R. Pedarsani, J. Walrand, and Y. Zhong, "Robust scheduling and congestion control for flexible queueing networks," in *Computing, Networking and Communications (ICNC), 2014 International Conference on*, pp. 467–471, IEEE, 2014.
- [21] R. Pedarsani, *Robust scheduling for queueing networks*. University of California, Berkeley, 2015.
- [22] R. Pedarsani, J. Walrand, and Y. Zhong, "Robust scheduling for flexible processing networks," *Advances in Applied Probability*, vol. 49, no. 2, pp. 603–628, 2017.
- [23] E. Lovisari, C. C. de Wit, and A. Kibangou, "Density/flow reconstruction via heterogeneous sources and optimal sensor placement in road networks," *Transportation Research Part C: Emerging Technologies*, vol. 69, pp. 451–476, 2016.
- [24] N. Mehr, J. Lioris, R. Horowitz, and R. Pedarsani, "Joint perimeter and signal control of urban traffic via network utility maximization," in *Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on*, IEEE, 2017.
- [25] E. Lovisari, C. C. de Wit, and A. Y. Kibangou, "Optimal sensor placement in road transportation networks using virtual variances," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pp. 2786–2791, IEEE, 2015.
- [26] X. J. 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*, vol. 19, no. 6, pp. 1133–1156, 2011.
- [27] M. Wright and R. Horowitz, "Fusing loop and gps probe measurements to estimate freeway density," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3577–3590, 2016.