Signal Control for Urban Traffic Networks with Unknown System Parameters

Negar Mehr[1], Jennie Lioris[2], Roberto Horowitz[1], and Ramtin Pedarsani[3]

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 and demands are known. However, it is envisaged that in the next generation of transportation networks with mixed autonomy, system parameters such as service rates may vary as autonomous vehicle penetration rate changes. Aligned with this, we propose a signal control strategy which, unlike previous approaches, can handle both unknown mean network demands and service rates. To this end, we use stochastic gradient projection to develop a cyclic iterative control, where at every cycle, the timing plan of the signals is updated. At each iteration, the update rule is based on the measured changes in the network queue lengths. If the network mean arrival and service rates are assumed to be constant, the proposed iterative signal control is guaranteed to converge to an optimal solution. 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 ongoing rise of traffic congestion has highlighted the importance of traffic control for every commuter. Efficient traffic control strategies can result in throughput increases and decreases in fuel consumption. In this work, we focus on the design of signal control strategies for urban networks. 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 fixed cycle time such that each set of nonconflicting phases at an intersection receives a fixed duration of green. A thorough analysis of fixed time control strategies is presented in [1].

Because of the simplicity of cyclic controls and their practicality, several methods have been proposed for determining the green durations that each stage must receive in a cyclic controller. Examples of such methods include SYNCHRO [2], VISGAOST [3], SCOOT [4], and OPAC [5]. SYNCHRO and VISGOAST used historical data to determine the signal timings in an offline fashion. In SCOOT and OPAC, the timing plan of each intersection was found such that a performance metric of upstream queues was optimized. A well known cyclic controller is traffic responsive control (TUC) which was proposed in [6], where a multivariable traffic–responsive feedback based regulatory control was proposed. Another type of signal control that was shown to be effective in stabilizing traffic networks called Max Pressure was proposed in [7], where the stage with the highest pressure was actuated, and the pressure for each stage was computed using the length of the neighboring queues.

A popular class of control strategies for signal control of urban and freeway networks is model predictive control (MPC) [8], [9], [10], [11]. Such controls are popular since they can systematically handle constraints. It is well known that in order for MPC 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 of synthesis tools in the formal methods community, temporal logic tools were used for the task of traffic control [12], [13]. The viability of such methods depends heavily on the existence of direct temporal specifications that describe the desired behavior of the system as well as accurate knowledge of system parameters.

A common feature of the majority of the aforementioned works is that they assume that network parameters such as service rates of queues and network 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 significant attention is going to be used for creating platoons of vehicles. It is shown that as the penetration rate of connected vehicles on the road varies, the service rates of network queues vary as well [14], [15]. This implies that the higher the penetration rate is, the higher the service rates will be. Hence, the previous approaches which assume that the network service rates are known to the controller may not be applicable for networks equipped with CVT. Therefore, it is important to come up with strategies that are robust to service rates. In addition to service rates, in practice, network demands might be unknown too. This further points to the necessity of developing controllers that are robust to the knowledge of network demands.

In this paper, we propose a signal control strategy that is robust to the knowledge of both network service rates and demands similar to [16], [17], [18]. We show how the framework of [18] which was first introduced for communication networks can be applied to traffic networks when network demands and service rates are unknown. We determine the green durations of every stage such that the control converges to the timing plan which is desired

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

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

[3]R. 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

for maximizing the network throughput. Our approach is different from the existing literature in that it *learns* the timing plans iteratively by measuring the changes in the queue lengths. In our approach, we still assume that turning ratios of the network are known, but this is not a restrictive assumption since turning ratios are normally available from surveys or counts [19]. We use the PointQ model [7] to formally define an urban network as a queuing system. We state the requirements that our synthesized control must satisfy in order to be implementable in a cyclic fashion. We describe how the green duration of each stage can be updated using a gradient projection algorithm such that all the flows in the network are balanced. Our iterative control scheme is guaranteed to converge to an optimal and "balanced" signal plan when the network parameters are constant. We further demonstrate the capability and performance of our algorithm in simulation environment.

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 a simulation example in Section V. We conclude the paper and discuss 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 elementwise 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. To distinguish matrices from vectors, we show matrices excursively with upper case letters. 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 the transpose of matrix X. 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 set $\mathcal{S}$.

III. NETWORK MODEL

In order to model the dynamics of a traffic network, we use PointQ model proposed in [7]. PointQ models traffic networks as store and forward queuing systems. It allows for easily characterizing feasible demand profiles and stabilizing signal controls. We let the network be denoted by $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of network nodes, and $\mathcal{L}$ is the set of network edges. The set of nodes $\mathcal{N}$ and edges $\mathcal{L}$ represent the sets of network intersections and links respectively. Let $|\mathcal{N}| = N$ and and $|\mathcal{L}| = L$ be the number of network nodes, and network links respectively.

In pointQ, the 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 or demands to the network. They are identified by the fact that the entry links do not have any starting node 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; thus, exit links do not have any end nodes in the network. We assume that the network demand and saturation flow rates are random variables. For each link $l \in \mathcal{L}$, 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 demand on link l respectively. Note that since we have assumed that exogenous arrivals enter the network only through entry links, $d_l = 0$, for all links $l \in \mathcal{L}_{\text{inter}} \cup \mathcal{L}_{\text{exit}}$.

At each node $n \in \mathcal{N}$, only certain movements or phases are allowed. For each such movement, a separate queue is considered in PointQ. 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 $l, m \in \mathcal{L}$, we use $r(l,m)$ to represent the fraction of vehicles that join link m when they leave link l, or equivalently, the probability of the event that a random vehicle joins link m when departing link l. It is assumed that $r(l,m)$'s are known a priori. Moreover, for each movement from link $l \in \mathcal{L}$ to link $m \in \mathcal{L}$, we use $\mu(l,m)$ to represent the mean saturation flow or service rate of the movement.

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

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

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

$$f_l = d_l, \qquad f(l,m) = r(l,m) f_l, \quad \text{if } l \in \mathcal{L}_{\text{entry}}, \tag{2}$$

$$f_l = \sum_{o \in \mathcal{L}} f(o,l), \quad f(o,l) = r(o,l) f_o, \quad \text{if } l \in \mathcal{L}_{\text{inter}} \cup \mathcal{L}_{\text{exit}}. \tag{3}$$

Now, we can proceed to how we define a cyclic control in our PointQ model.

A. Cyclic Control

Assume that all actuators of the signalized intersections are cyclic, i.e. they operate on a common cycle time T. For such controllers, at each node $n \in \mathcal{N}$, there exist multiple stages $s^n_j, 1 \leq j \leq S^n$, where S^n is the total number of stages at node n. Each stage s^n_j is a set of nonconflicting movements that can be actuated simultaneously. Assume that stage s^n_j receives green for g^n_j fraction of the cycle time. A single stage might contain several movements. Additionally, as a movement might be actuated during multiple stages, we need to define the fraction of the cycle time that a movement receives green when a stage s^n_j is actuated. For a movement from link l to m belonging to stage s^n_j, we use $g^n_j(l,m)$ to denote the fraction of green that this movement receives during stage s^n_j. Clearly, if multiple movements are actuated during a stage s^n_j, the green durations that they receive during stage s^n_j 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 the same stage s^n_j, we must have

$$g^n_j(l,m) = g^n_j(u,v) = g^n_j, \quad \forall\, (l,m) \text{ and } (u,v) \in s^n_j. \tag{4}$$

Fig. 1. Schematic of an intersection.

Note that at each node $n \in \mathcal{N}$, stage green durations g_j^n must add up to 1; therefore, we must have

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

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

As a movement might receive green during multiple stages, for each movement from l to m, we use $p(l,m)$ to be the fraction of the aggregate green durations that the movement receives during a cycle

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

Example: To illustrate the notation, consider the intersection shown in Figure 1. There is only one node in this network. Let this node be indexed by 1. Thus, superscript 1 is considered for the single node of the network. 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. Assuming that there exist only through and right movements, there are eight queues in the network, where the origin–destination links for all network queues are: (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 stage lasts half of the cycle time. Assume that the following movements are actuated during each stage.
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

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

while, for the second stage, we have

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

Using (5), we have

$$g_1^1 + g_2^1 = 1.$$

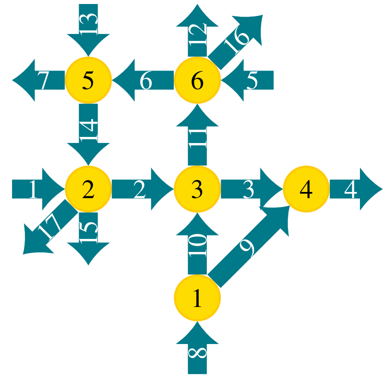

Fig. 2. The topology of the example network.

Finally, the aggregate green durations of the movements are obtained 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, we introduce a compact notation of the model parameters and variables. We use a notation similar to the one adopted in [20]. 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:

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

In addition to the vector of *link* flows, we can construct the vector of average *movement* flows. Assume that there exits a total of B possible movements in the network. We let $\varphi \in \mathbb{R}_+^B$ be the vector of movement flows $f(l,m)$, for all $l,m \in \mathcal{L}$ such that movement from link l to link m is allowed. Moreover, we collect the aggregate fractions of green that the network movements receive, $p(l,m)$'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, where 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 $\Gamma \in \mathbb{R}^{B\times L}$

$$\varphi = \Gamma f, \tag{8}$$

where the matrix Γ is such that at j_{th} row of Γ, 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

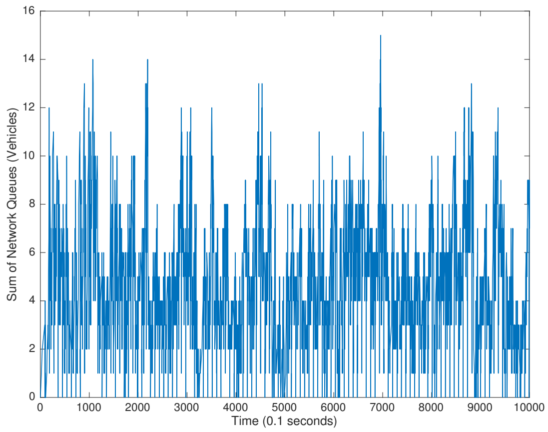

Fig. 3. Sum of all queues vs. time.

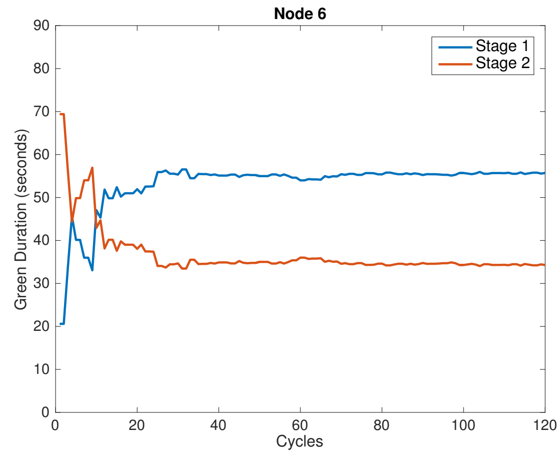

Fig. 4. Green duration of stages in node 6.

j_{th} movement. A given demand pattern is called feasible if and only if the mean demand vector d is such that for every $l, m \in \mathcal{L}$, we have

$$f(l,m) < \mu(l,m)p(l,m).$$

Next, to represent conditions (4), (5), and (6), we collect fractions of green that network 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 such durations. Then, we can rewrite Equation (6) for all queues as:

$$p = A_{g\to p}\, g, \tag{9}$$

where $A_{g\to p}$ is the linear mapping of appropriate dimension. Moreover, we define the linear mapping A_{eq} to encode (4) for all pairs of queues that belong to the same stage of an intersection,

$$A_{\text{eq}}\, g = \mathbf{0}. \tag{10}$$

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

$$A_{\text{sum}}\, g = \mathbf{1}^{N\times 1}. \tag{11}$$

Thus, the requirements imposed for cyclic implementation of a timing plan are represented by Equations (9), (10), and (11).

Finally, we need to define the vector of queue lengths for all network queues. In the remainder of the paper, we let $q \in \mathbb{R}^B$ 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 define the convex set $\mathcal{C}$ to be the following

$$\begin{aligned}\mathcal{C} = \{p \in \mathbb{R}^B \mid \exists\, g \geq 0, \text{ such that } p = A_{g\to p}\, g,\\ A_{\text{eq}}\, g = 0,\ A_{\text{sum}}\, g = 1\}.\end{aligned} \tag{12}$$

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 (9), (10), and (11). We use k to denote the cycle or time index. Every time step of the controller is assumed to last a cycle time T. We let p_k and g_k be the aggregate green durations and stage green durations of network movements during cycle k. We let $E_k \in \mathbb{R}^{B\times B}$ be a diagonal matrix with each diagonal entry e_{ii} being 1 if the i_{th} queue is nonempty at the beginning of the k_{th} cycle and zero otherwise. We further let q_k be the vector of queue lengths for all network queues at the beginning of cycle k. We also define the sequence of step sizes or learning rates $\{\beta_k\}$ to be a decreasing sequence such that

$$\begin{aligned}\beta_k \to 0 \ \text{ as } k \to \infty,\\ \sum_{k=0}^{\infty} \beta_k = \infty,\\ \sum_{k=0}^{\infty} \beta_k^2 < \infty.\end{aligned} \tag{13}$$

The above conditions on β_k are standard assumptions required for the convergence of stochastic gradient projection algorithms. Examples of such sequences include $\beta_k = \frac{1}{k}$, which satisfies the above conditions. We also use $\Delta q_k = q_k - q_{k-1}$ to denote the vector of the difference in queue lengths at each time step k, where $k \geq 1$. Finally, we define matrix $\Lambda \in \mathbb{R}^{L\times B}$ to be such that at each row $i \in L$, all elements of the matrix are zero except for the elements located at j_{th} columns with j being the index of the queues that originated from 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 indices of the queues for 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 similarly.

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

1) Initialize p_0 with an arbitrary feasible value (such that

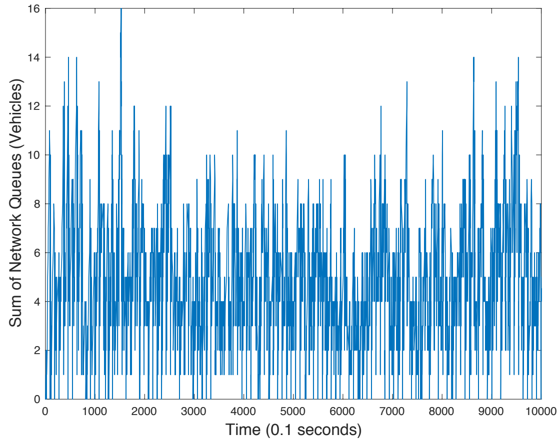

Fig. 5. Sum of all queues as a function of time when the service rates were varied.

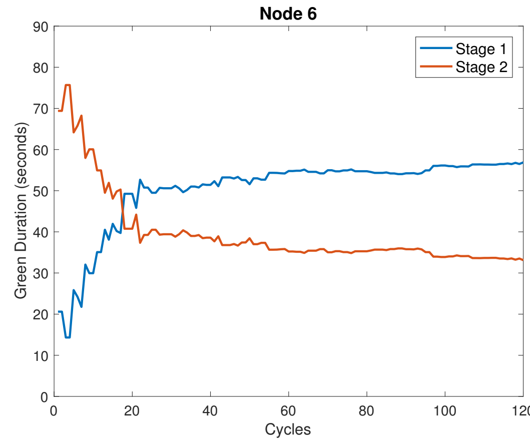

Fig. 6. Green duration of the stages in node 6 when the service rates are varied.

$p_0 \in \mathcal{C}$).

2) At each time step $k, k \geq 1$, update vector p by

$$p_k = \left[p_{k-1} + \beta_k \left(\Gamma E_k (I - R^T)^{-1} \Lambda \Delta q_k\right)\right]_{\mathcal{C}}, \quad (14)$$

where $[.]_{\mathcal{C}}$ is the convex 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 Δq_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 other words, once p_k is found, a vector g_k such that $p_k = A_{g\to p} g_k$ is required for implementing the new timing plan. Since there might be multiple vectors g_k such that $p_k = A_{g\to p} g_k$, we can obtain a vector of stage green durations g_k by simply solving a least squares problem for an updated p_k.

$$g_k = \min_x \; \|p_k - A_{g\to p} x\|^2 .$$

Now, we explain the intuition behind our algorithm. If the vector of demand mean values d were known, then, the vector of movement flows φ would have been easily obtained by

$$\varphi = \Gamma (I - R^T)^{-1} d.$$

If the matrix of service rates M were also known, then, the gradient projection update rules of the form

$$p_k = [p_{k-1} - \beta_{k-1}(\Gamma f_{k-1} - M p_{k-1})]_{\mathcal{C}}$$

would have solved the following optimization problem:

$$\begin{aligned} \underset{p}{\text{minimize}} \quad & \frac{1}{2}\|\varphi - Mp\|^2 \\ \text{subject to} \quad & p \in \mathcal{C}. \end{aligned} \quad (15)$$

Note that the solution of the optimization problem (15) yields the optimal signal plan that would have balanced all the flows in the network while guaranteeing rate stability of the queues. However, when service rates M and demands λ are unknown, the vector of movement flows φ is not available. Consequently, we use Δq_k to estimate $(\Gamma f - M p_k)$ as Δq_k is indeed an unbiased estimator of the gradient term with finite variance. Therefore, our update rule in (14) is a stochastic gradient projection algorithm for a convex optimization problem, which is guaranteed to converge for appropriate choices of the step size in (13). Moreover, as p_k converges to the optimal signal plan p^* that is the solution of (15), one can show that the network queues remain stable using similar proof ideas as the one in [18]. The framework we have used in this work was first introduced in [18] for communication networks. In this work, we have shown how the framework can be employed for traffic networks; therefore, we do not repeat the proofs on convergence of the algorithm and its optimality. Interested reader is referred to [16], [18]. It is also important to mention that the convergence was formally guaranteed for stationary demand and service rates. However, in practice, it has been observed that as long as the changes in the network demand and service rates are slow enough, the algorithm can learn to adapt itself to the new set of system parameters.

It is noteworthy that, in general, queue length measurement might not be available for all queues in the network. But, queue estimation methods have been proposed in the literature using loop detectors, GPS data or combination of the two for queue estimation [21]. Hence, when the queue lengths are not available, such methods can be utilized for estimating queue lengths and Δq_k as a result.

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 $T = 90$ seconds. All nodes have cyclic controllers with known stages. Nodes 2,3,5 and 6 have 2 stages, while nodes 1 and 4 are not signalized. 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 were unknown to the controller. For the simulation purpose, we used a typical set of demand profiles with feasible Poisson arrivals with fixed mean value. We used pointQ simulator [22] for our analysis.

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 was not growing and remained bounded). In addition to network stability, the convergence of the network timing plans was also achieved as shown in Figure 4. Figure 4 shows the convergence of the stage green durations for node 6. Using Figure 4, we observe that the control algorithm learned a stabilizing timing plan over time despite the arbitrarily provided initial timing plan. Moreover, the convergence occurred quite fast, around 30_{th} 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.

In order to verify the practicality of the proposed control algorithm, we further repeated the simulation but decreased service rates by 20% at the 20_{th} cycle to investigate how the control adapted to the new changes. As shown in Figure 5, since the demand was still feasible with the reduced service rates, the controller could successfully maintain the stability of the system queues. Figure 6 demonstrates the green durations of the two stages at node 6. The convergence of the green durations was achieved in this case too. During our simulations, we found that convergence and stability were achieved for other feasible demands too that we omit reporting due to the lack of space and similarity of behavior.

VI. Conclusion and future work

In conclusion, we proposed an iterative cyclic signal control for network of signalized intersections such that the control is unaware of the network mean demands and service rates. Our control is iterative in the sense that at the beginning of every cycle time, it decides on the green durations based on the measured changes in the lengths of queues. We demonstrated though 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, 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 controller can be decentralized.

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] C. Diakaki, M. Papageorgiou, and T. McLean, "Integrated traffic-responsive urban corridor control strategy in glasgow, scotland: Application and evaluation," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1727, pp. 101–111, 2000.

[7] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.

[8] 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.

[9] 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.

[10] 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.

[11] N. Mehr and R. Horowitz, "Probabilistic freeway ramp metering," *rN*, vol. 1, no. f1, p. 1, 2016.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] R. Pedarsani, *Robust scheduling for queueing networks*. University of California, Berkeley, 2015.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] J. Lioris, A. A. Kurzhanskiy, and P. Varaiya, "Control experiments for a network of signalized intersections using the'. q'simulator.," in *WODES*, pp. 332–337, 2014.