2018 21st International Conference on Intelligent Transportation Systems (ITSC)
Maui, Hawaii, USA, November 4-7, 2018

Koopman Operator Approach for Instability Detection and Mitigation in Signalized Traffic

[1]Esther Ling, [2]Lillian Ratliff and [1]Samuel Coogan

***Abstract*—The objective of this paper is to demonstrate an application of the Koopman Operator approach for automated detection of unstable behavior in traffic dynamics. We propose an algorithm that searches for long sequences of unstable eigenvalues in the learned dynamics as an anomaly feature, and demonstrate feasibility of the approach on a case study day with abnormally high queue lengths due to an accident. Secondly, we present a method for modeling traffic dynamics with signal phases included as exogenous input. Given an unusually congested condition, the analysis is consistent in that longer green times for the affected leg result in less severe congestion and vice versa for shorter green times. This shows promise for anticipating queue behavior under modified traffic signal phase sequences. We encapsulate these ideas within the larger picture of building improved adaptive traffic control systems.**

I. Introduction

At a traffic intersection, the dynamics of the queue charge-discharge cycle is largely governed by the stochastic arrival of vehicles and exogenous forcing from traffic signaling. These dynamics are highly oscillatory and nonlinear [1]. An approach that is becoming increasingly common to modeling such nonlinear behavior in other domains is not by enforcing priors, but by relying on collected data. To this end, we propose to model traffic dynamics using dynamic mode decomposition (DMD), an equation-free method. Dynamic mode decomposition has been connected to Koopman Operator theory [2], [3], which provides an operator-theoretic view of nonlinear systems through the evolution of functions on the state, also known as observables. Nonlinear systems have been traditionally analyzed according to the differential equations based upon the laws of physics, or by evolution of states, in the Poincaré geometric formulation. However, these approaches are ill-conditioned when the number of states grow exponentially large [2]. In the transformed coordinates postulated by Koopman [4] in 1931, the observables evolve according to a linear law. The shift in focus ushered in by the Koopman Operator can arguably be classified under the larger branch of "algorithmic modeling" as described in [5] and [6], which provide an exposition on the development of applied mathematics for nonlinear dynamical systems.

*This research was supported in part by the National Science Foundation under awards 1749357 and 1736582

[1]E. Ling lingesther@gatech.edu and S. Coogan sam.coogan@gatech.edu are with the School of Electrical and Computer Engineering, Georgia Institute of Technology

[2]L. Ratliff ratliffl@uw.edu is with the Department of Electrical Engineering, University of Washington

The Koopman Operator framework has proven successful in other domains such as data-driven modeling of power systems [7], financial markets [8] and disease [9]. In our current context of signalized traffic, Koopman Operator provides a natural framework for defining abnormal events in signalized traffic in terms of *instability* and identifying such instabilities through spectral analysis.

Increasingly, more cities are seeking to adopt adaptive traffic control systems (ATCS). Traditional traffic systems use a fixed timing plan with different cycles to cater for different times of the day, based on historical traffic flows. Though this works to ensure eventual traffic flow in the long-term, it fails to accommodate short-term fluctuations rampant during peak hours and sudden surges during a traffic incident.

Nonetheless, the performance of an adaptive traffic control system is largely dependent on its ability to accurately sense vehicles and optimize signal timings in real-time. In a typical ACTS system, a modified timing plan will be chosen from a library of phase-splits in response to a change in traffic condition. To verify phase-splits, a software simulator such as Synchro is used where traffic is normally modeled according to a Poisson process. This approach however, is dependent on how good of an approximation the model is to the true intersection. Furthermore, abnormal behavior is difficult to model, being by nature, rare events and thus manifest themselves in a wide variety of types and times.

Moreover, as a precursor to having a smart traffic system, more municipalities are seeing investments placed into traffic sensors in order to facilitate real-time detection of vehicles. The installation of such sensors has resulted in the collection of large volumes of historical data, which begs the use of creating more data-driven informed traffic models.

To make concrete the focus of this paper, we home in on a sub-problem of the adaptive traffic control system: queue breakdown detection and modified phase-split evaluation to anticipate queue behavior. Our approach emphasizes use of data collected from traffic sensors. In particular, we propose a method for early detection of queue breakdown and use DMD to model queue dynamics with signal phases as exogenous input.

The remainder of the paper is organized as follows: Section II provides a brief review on Koopman Operator theory and dynamic mode decomposition. Section III describes the data used in our study, and Section IV presents our algorithm for instability detection and queue mitigation. Finally, we provide concluding remarks in Section V.

978-1-7281-0322-8/18/$31.00 ©2018 IEEE

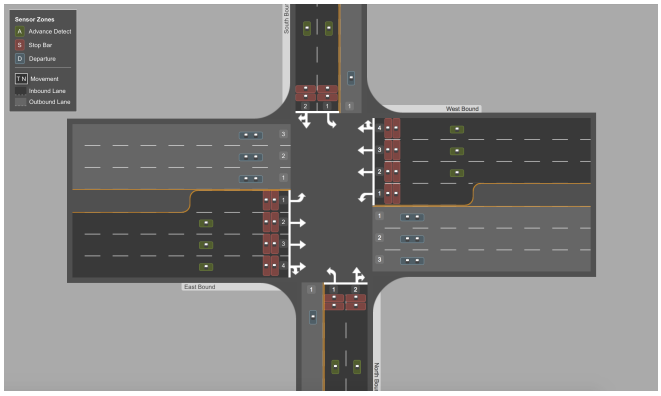


Fig. 1. Montrose Rd & Tildenwood Intersection. Note the positioning of Legs, Lanes, Stopbar Detectors (in red), Advanced Detectors (in green) and Departure Detectors (in blue). Figure courtesy of Sensys Networks, Inc.

II. Koopman Operator Background

Consider a nonlinear discrete-time system

$$\begin{aligned} x_{k+1} &= \mathcal{T}(x_k) \\ y_k &= f(x_k) \end{aligned} \tag{1}$$

where $x_k \in \mathbb{R}^D$ is a state vector, $\mathcal{T}$ is a nonlinear vector map that describes the evolution of state trajectories, $y_k \in \mathbb{C}^M$ is the measured output, and f is the observation of the state, also known as the output function that maps states to measurements such that $f : \mathbb{R}^D \to \mathbb{C}^M$, and k is the time-step.

The Koopman Operator $\mathcal{K}$ is the infinite-dimensional operator that advances the observed quantities at each step by

$$(\mathcal{K}f)x_k = f(x_{k+1}) \; . \tag{2}$$

Dynamic mode decomposition is a data-driven algorithm that provides a finite-dimensional approximation of $\mathcal{K}$. Developed within the fluid dynamics community [10], connections were later made as a data-driven method for approximating the modes of the Koopman Operator [3]. See [2] for a thorough review.

DMD is an equation-free method in that it does not enforce prior belief on $\mathcal{T}$ and f [2], [7]. Instead, it utilizes data snapshots of the system sampled at intervals of Δt to learn a locally-linear model described by the proxy matrix A so that

$$x_{k+1} \approx Ax_k \tag{3}$$

where $A \in \mathbb{R}^{M \times M}$. A is obtained by solving the optimization program

$$\min_A ||X_2 - AX_1||_F^2 \tag{4}$$

where $||.||_F$ is the Frobenius norm and X_1 and X_2 are time-shifted data matrices of dimension $\mathbb{R}^{M \times (N-1)}$:

$$X_1 = \begin{bmatrix} | & & | \\ x_1 & \dots & x_{N-1} \\ | & & | \end{bmatrix} \; X_2 = \begin{bmatrix} | & & | \\ x_2 & \dots & x_N \\ | & & | \end{bmatrix} . \tag{5}$$

where N is the total number of snapshots.

(4) has a closed form solution that employs the singular value decomposition (SVD) of X_1:

$$\begin{aligned} A &= X_2 X_1^\dagger \\ &= X_2 V \Sigma^{-1} U^T \end{aligned} \tag{6}$$

where $\dagger$ indicates the pseudo-inverse. Since $\text{rank}(X_1) = r \leq \min(M, N-1)$, we have $U \in \mathbb{R}^{M \times r}$, $\Sigma \in \mathbb{R}^{r \times r}$, $V \in \mathbb{R}^{(N-1) \times r}$. If M is large, computing the eigen-decomposition of A directly is compute-intensive. To bypass this, DMD approximates the eigen-decomposition of a rank-reduced $\tilde{A}$

$$\begin{aligned} \tilde{A} &= U^T A U \\ &= U^T X_2 V \Sigma^{-1} \end{aligned} \tag{7}$$

where $\tilde{A} \in \mathbb{R}^{r \times r}$.

Further computational gains can be made if there is low rank structure in the dynamics. The number of singular values in the SVD of X_1 can be truncated to $\tilde{r} \leq r$:

$$\tilde{A} = \tilde{U}^T X_2 \tilde{V} \tilde{\Sigma}^{-1} \tag{8}$$

where $\tilde{U} \in \mathbb{R}^{M \times \tilde{r}}$, $\tilde{\Sigma} \in \mathbb{R}^{\tilde{r} \times \tilde{r}}$ and $\tilde{V} \in \mathbb{R}^{(N-1) \times \tilde{r}}$. A rule of thumb for choosing $\tilde{r}$ is by looking at the abruptness of decay of the singular values.

In [11], it is shown that the eigen-decomposition $\tilde{A}W = W\Lambda$ approximates the eigen-decomposition of A where Λ, a diagonal matrix of eigenvalues are the eigenvalues of A, and the eigenvectors of A, termed "DMD modes" are approximated by

$$\Psi = X_2 \tilde{V} \tilde{\Sigma}^{-1} W. \tag{9}$$

Each DMD mode ψ_i, which is a column in Ψ, is associated with a growth/decay and/or oscillatory component determined by the eigenvalue λ_i. Moreover, $|\lambda_i|$ provides stability properties of each mode so that if $|\lambda_i| < 1$, then the corresponding DMD mode is stable. This property was utilized in [12], which used the number density of unstable eigenvalues to analyze cascading outages of the 2006 system disturbance of the European Power Grid.

In the event that the number of snapshots is much larger than the spatial dimension, i.e. $N \gg M$ and X_1 and X_2 are wide matrices, using the original DMD formulation does not recover the dynamics appropriately [11], [13]. This is the case when considering traffic at a high-level, such as at intersection level. One common method to increase the rank is to append h time-shifted observations to the data matrices [14]:

$$\tilde{X}_1 = \begin{bmatrix} | & & | \\ x_1 & \dots & x_{N-h} \\ | & & | \\ x_2 & \dots & x_{N-h+1} \\ | & & | \\ \vdots & & \vdots \\ | & & | \\ x_h & \dots & x_{N-1} \\ | & & | \end{bmatrix} \; \tilde{X}_2 = \begin{bmatrix} | & & | \\ x_2 & \dots & x_{N-h+1} \\ | & & | \\ x_3 & \dots & x_{N-h+2} \\ | & & | \\ \vdots & & \vdots \\ | & & | \\ x_{h+1} & \dots & x_N \\ | & & | \end{bmatrix} \tag{10}$$

TABLE I
TIMING PLAN FOR THE INTERSECTION. (1-EBLT, 2-EB, 4-SB, 5-WBLT, 6-WB, 8-NB). (LEFT) 9.30AM-3.30PM, 110S. (RIGHT) 3.30PM-7.00PM, 120S. PHASES 1, 2, 5 AND 6 ARE IN BARRIER 1, WHILE PHASES 4 AND 8 ARE IN BARRIER 2.

Phases	1, 5	2, 6	4, 8	Phase	1, 5	2, 6	4, 8
Time (s)	16	49	45	Time (s)	23	53	44

where $\tilde{X}_1$ and $\tilde{X}_2$ are Hankel matrices. This method is also commonly applied in subspace identification techniques.

In [15], DMD is extended to model an exogenous control input, known as dynamic mode decomposition with control (DMDc):

$$x_{k+1} \approx Ax_k + Bu_k \tag{11}$$

where $u_k \in \mathbb{R}^Q$ is the control action.

By collecting the inputs column-wise into a matrix $\mathbb{U}$,

$$\mathbb{U} = \begin{bmatrix} | & & | \\ u_1 & \dots & u_{N-1} \\ | & & | \end{bmatrix}, \tag{12}$$

A and B can be approximated using matrix augmentation. In particular,

$$\begin{aligned} X_2 &= AX_1 + B\mathbb{U} \\ &= \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} X_1 \\ \mathbb{U} \end{bmatrix} \\ &= G\Omega \end{aligned} \tag{13}$$

where $G \in \mathbb{R}^{M \times (M+Q)}$. Then, G can be computed as follows:

$$\begin{aligned} G &= X_2\Omega^\dagger \\ &= X_2\hat{V}\hat{\Sigma}^{-1}\hat{U}^T \end{aligned} \tag{14}$$

where $\hat{V}$, $\hat{\Sigma}$ and $\hat{U}$ are from the SVD of Ω with $\Omega = \hat{U}\hat{\Sigma}\hat{V}^T$.

Since the first M components in the left singular vectors of $\hat{U}$ come from A, while the remaining Q components are from B, A and B can be determined from G by extracting the appropriate rows in $\hat{U}$ according to

$$\begin{aligned} A &\approx X_2\hat{V}\hat{\Sigma}^{-1}\hat{U}_1^T \\ B &\approx X_2\hat{V}\hat{\Sigma}^{-1}\hat{U}_2^T \end{aligned} \tag{15}$$

where $\hat{U}_1 \in \mathbb{R}^{M \times r}$, $\hat{U}_2 \in \mathbb{R}^{Q \times r}$ and $\hat{U} = \begin{bmatrix} \hat{U}_1 \\ \hat{U}_2 \end{bmatrix}$.

We note that it is also possible to append time-delayed rows to $\mathbb{U}$. This effectively takes into account knowledge of past signal phases in learning the traffic dynamics.

III. DATA OVERVIEW

The experiments in this paper make use of high resolution data from the Montrose Rd & Tildenwood intersection. This intersection is part of an arterial network in Montgomery County, Maryland. In the intersection, there are 4 legs designating direction: East-Bound (EB), West-Bound (WB), South-Bound (SB) and North-Bound (NB). At each of these legs, sensors have been installed to detect arrival and departure of vehicles entering a particular intersection.

Fig. 2. Signal phase events from 2.30pm-4.00pm on 10 February 2017. The color of each column corresponds to the state of the traffic light.

In particular, Advanced Detectors are positioned 200-300 feet upstream of the traffic light to detect vehicles entering the leg. Stopbar Detectors on the other hand are positioned at the departure of the traffic light to detect vehicles exiting the leg. Fig. 1 shows the Advanced Detector and Stopbar Detector placements. The data is obtained courtesy of Sensys Networks, Inc.

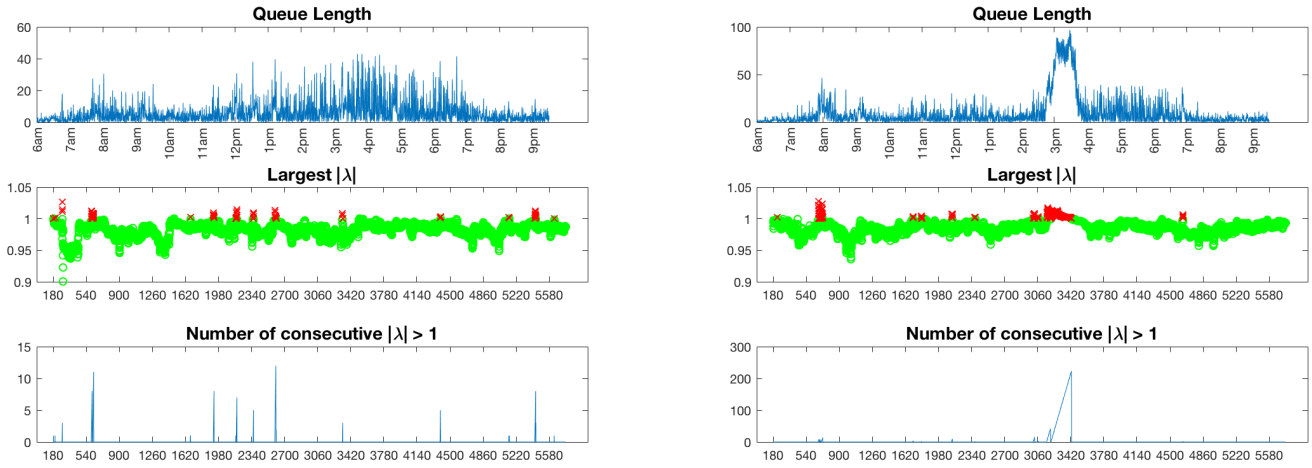
A time-stamped event is logged every time a vehicle crosses either of the sensors described above. Data captured with the logged event relevant to our use is current phase during the detection (red / yellow / green) and time to end of phase (s). Using the detection event data, we obtain queue length estimation by incrementing the queue length for each Advanced Detector event and decrementing for each Stopbar Detector event. To accommodate downward and upward bias, we employ a similar method as in [16].

Also available are signal phase events from the actuation of the traffic lights during the course of the day. This event data provides the actual phase-split lengths within a cycle. The green times for each phase may vary within a cycle depending on flow across all legs. For example, legs that experience low vehicle flow may have a shorter green allocation for that particular cycle. We provide an example of the signal phase event data in Fig. 2, where from time-steps 0-3600, the cycle is 110s, while from time-step 3600 onwards, the cycle is 120s. Notice that there is some variation in the length of the green times in the 110s cycle. The event data is logged by a controller conflict monitoring card, time synchronous to 0.1 seconds and is unevenly spaced. To decrease computation time and create evenly spaced samples, we down-sample the data to 1s. One difficulty we encounter is how to down-sample the data meaningfully. We find that the most reasonable approach is to copy over the most recent phase where a data point does not exist.

IV. EXPERIMENTS

A. Instability Detection

In this section, we demonstrate an algorithm for identifying unstable dynamics in queue lengths. In an adaptive traffic system, it is advantageous to anticipate congestion before it



(a) West-Bound Leg, Normal Day

(b) West-Bound Leg, Accident Day

Fig. 3. (Top Subplots) Queue Lengths. (Middle Subplots) $|\lambda_{1,k}|$ (unstable $\lambda_{1,k}$ are shown in red, while stable $\lambda_{1,k}$ are green). (Bottom Subplots) Υ_k. Each data-point in the Middle and Bottom subplots corresponds to the dynamics of the previous 180 data samples. Notice that the number of consecutive unstable eigenvalues is much larger during the accident.

Algorithm 1 Instability Detection Algorithm

Initialize $ctr = 0$ and $N = 180$.
for $k = N : \tilde{N}$ **do**
  Run DMD to learn A using observations from $k - N$ to k.
  if $|\lambda_{1,k}| > 1$ **then**
    $ctr = ctr + 1$
  else
    $ctr = 0$
  end if
  $\Upsilon_k = ctr$.
  if $\Upsilon_k > \epsilon$ **then**
    Flag an abnormal event condition.
  end if
end for

happens, so that control can be exerted early to alleviate the event. For example, while presence of high queue volumes is indicative of a traffic incident, detecting sustained queue growth would serve as an early warning. To this end, presence of an unstable eigenvalue in the learned matrix A, i.e when A contains an eigenvalue λ such that $|\lambda| > 1$ can be used to identify growth. Then, learning A in a rolling window of queue measurements can be used to detect a sequence of growth conditions.

To formulate the problem: let the snapshot vector x_k be the queue length at a particular leg, with each increment of time-step k corresponding to $\Delta t = 10$ s. Then, form the data matrices $\tilde{X}_1$ and $\tilde{X}_2$ according to (10), using $h = 10$. Then, run DMD in a rolling window fashion with a stride of k, looking back at the N most recent samples. Apply a rank truncation of $\tilde{r} = 10$. Also, let $\lambda_{1,k}$ be the largest eigenvalue of A at time k, Υ_k the count of consecutive $|\lambda_1| > 1$ at time k, and $\tilde{N}$ the total number of samples considered in the day. Alg. 1 summarizes this process.

Our investigation day is Friday, 10 February 2017. At approximately 2.47pm, there was a reported accident at the Tildenwood-Montrose Road Intersection which affected the WB and EB movements. We provide a plot of the queues at the WB movement, shown in the top subplot in Fig. 3b. (The accident data is obtained from the Maryland Open Data Portal [17]). Notice the prolonged spike from the time of the accident until about 4pm, which is more severe compared to the peak hour in the morning. For comparison, we also provide queue lengths at the same movement on a normal Friday, 17 February 2017 in the top subplot in Fig. 3a.

Using Alg. 1, we compare the number of unstable eigenvalues for the normal day and the accident day. We choose $N = 180$, which corresponds to 30 minutes of data binned at 10s intervals. Choice of N is related to how long of a window to model the dynamics. Since accidents may have sustained growth compared to peak hour growth, a fairly long window such as 30 minutes would be appropriate.

In both Fig. 3a and Fig. 3b, the top subplot shows queue lengths at the WB leg, the middle subplot shows $|\lambda_1|$ obtained at each time-step, where unstable λ_1 are shown in red while stable λ_1 are green, and the bottom subplot shows Υ_k. Notice that the number of consecutive unstable eigenvalues is at least five times larger during the accident compared to normal peak hours. At $k = 3420$, $\Upsilon_{3420} \approx 200$ in Fig. 3b. To provide intuition on how to interpret Υ_k: since a stride of $10s$ is used, this corresponds to about 33 minutes of sustained queue growth as of time-step $k = 3420$.

Upon inspection of Fig. 3a, Υ_k does not exceed 15 for regular peak-hour traffic, thus, choosing $\epsilon > 15$ may be appropriate for this leg. Using this incident flagging strategy, a change in signal timing could have been effected earlier to alleviate the congestion such as at time-step 3200 (Fig. 3b), instead of waiting till time-step 3420.

Given that different intersections have different queue capacities dependent on the number of lanes, as well as varying traffic volume and patterns, ϵ would be location-

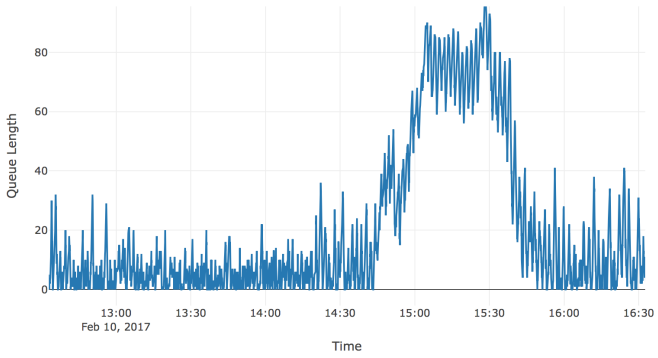


Fig. 4. Zoomed-in queue during the accident at the West-Bound leg. Notice the queue starts to clear at 3.30pm.

specific. To set a robust threshold parameter ϵ in Alg. 1, we suggest studying historical traffic flow and typical growth patterns. Too low of a value may result in false alarms. The objective also matters; if the aim is to distinguish between peak-hours and congestion worse than peak-hour traffic, then ϵ would need to be slightly larger than growth lengths during peak-hours. On the other hand, if the goal is to highlight when congestion exceeds a particular value, then ϵ would be set differently. For example, if a particular leg does not typically experience queue growth for longer than 10 minutes, then $\epsilon > 60$ would be a possible choice.

B. Queue Mitigation

In this subsection, we are interested in evaluating anticipated queue lengths under a different timing scheme than what was used during the accident. An accident may occur during any time of the day, necessitating adaptive changes to accommodate queue build-up. Suppose that the accident occurs at an off-peak hour. We then expect that not all the legs would be affected, and so providing an extended green time at the accident leg would facilitate faster clearing of the queue while ensuring that the non-affected legs are not negatively affected by the new phase-split. This is the case here, where only the WB and EB legs are affected by the accident. These legs belong to the same barrier (a barrier is used to group non-conflicting movements), and thus have the same green time. The timing plan for this intersection (courtesy of MCDOT) is shown in Table I).

Algorithm 2 Multi-Step Reconstruction

1: Inputs: A, B, $\{u_1, ..., u_N\}$ and x_1.
2: Returns: $\{x_2, ..., x_N\}$
3: **for** $k = 1 : N$ **do**
4: $\quad x_{k+1} = Ax_k + Bu_k$
5: **end for**

Furthermore, notice that in Fig. 4, the queues formed from the accident start to clear at 3.30pm. Interestingly, we note that the signal phase happens to have a scheduled change from the 110s cycle to 120s cycle at 3.30pm (see Table I). What is additionally interesting is that in the 120s cycle, the green times of the WB-EB phases are extended. We are curious if the signal phase had any effect in clearing the queue, and not due to fewer vehicles entering the queue at 3.30pm. We utilize DMDc to carry out this investigation.

We set up the problem as follows. Define the input vector u_k at time k as the signal phase at each turn movement at the affected intersection i.e.,

$$u_k = \begin{bmatrix} u_k(1) \\ u_k(2) \\ \vdots \\ u_k(12) \end{bmatrix}$$

where

$$u_k(i) = \begin{cases} 0, & \text{if } u_k(i) = \text{red} \\ 1, & \text{if } u_k(i) = \text{green or yellow} \end{cases}$$

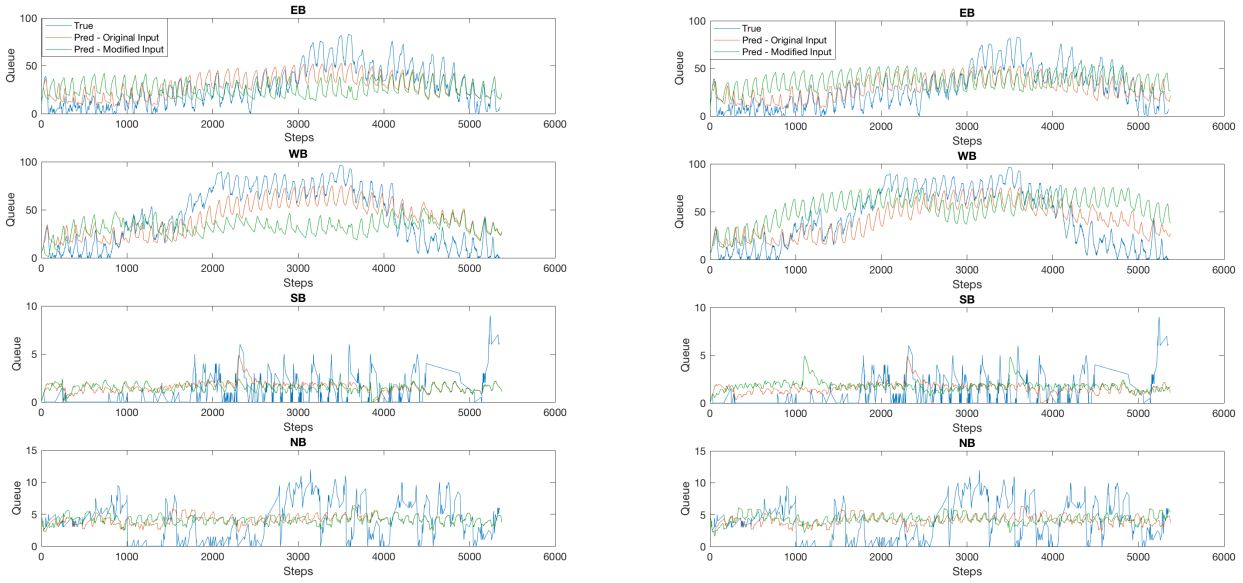
Let the state vector x_k consist of aggregated queue lengths of each leg at the intersection so that $x_k \in \mathbb{R}^4$. Then, form the data matrices $\tilde{X}_1$, $\tilde{X}_2$ and $\mathbb{U}$ respectively, as in (10) and (12), using $h = 12$. We use a rank truncation $\tilde{r}$ based on the number of singular values greater than $1e^{-10}$.

First, we learn the A and B matrices using (15) where u_k consists of the original signal phase. x_1 is the set of queues at 2.30pm, x_N is the set of queues at 4pm, and similarly for u_1 and u_N. To validate that A and B encode an appropriate model of the queue dynamics, we reconstruct the queues in multi-step fashion using x_1 as the initial condition and the original $\{u_1, ..., u_N\}$ as in Alg. 2. The reconstructed queues for the EB and WB legs (shown in red) in Fig. 5 show a good approximation of the dynamics of the original queues (shown in blue).

Next, we investigate the effect of modified inputs for two cases: (i) longer green-splits for the WB-EB phases, and (ii) shorter green-splits for the WB-EB phases. For the first case, $\{u_1, ..., u_N\}$ consists of signal phases from 3.30pm-4.30pm, which corresponds to the 120s cycle when the green-splits for the WB-EB legs are extended. In this period, the WB-EB phases are green for $60 - 70\%$ of each cycle period on average. To obtain an appropriate range of inputs from the signal phase data for the second case, we refer to Fig. 2. In this figure, the signal phases from time-steps 1400-3600 (corresponding roughly to 2.50pm-3.30pm) reflect the programmed phase-splits when there is sufficient flow from all legs, where the WB-EB green times are reduced. We build $\{u_1, ..., u_N\}$ using signal phases from this period. The WB-EB green-times are reduced to $45 - 60\%$ in this period.

For each of the cases, we again reconstruct the queues using Alg. 2 with the original x_1 as the initial condition. With the modified signal phase under longer green-split, the queues for the WB-EB legs are less severe (Fig. 5a). Conversely, under the shorter green-split, there is persisted high volume of queues at the WB-EB legs (Fig. 5b). The SB and NB queues, having much lower and less consistent vehicle flow, are not adversely affected by the modified splits.

These results are consistent with the expectation that the effect of extended green times for the relevant leg during an accident ought to result in faster clearing of queues,



(a) Longer green-split for WB-EB phases

(b) Shorter green-split for WB-EB phases

Fig. 5. DMDc to simulate effect of modified inputs on all legs. (Left) Under the extended green times for the WB and EB legs, the queues are less severe. (Right) Conversely, predicted queues under the shorter green-split show sustained congestion.

while a shorter allocation of green time results in persisted high queue volumes. The analysis suggests that DMDc may be a suitable tool for verifying queue dynamics to plan an appropriate green-split during abnormal traffic conditions.

V. Conclusion

We have demonstrated a method for detecting abnormal instabilities at a traffic intersection. Sustained growth of queues is undesirable and early detection can aid in serving mitigative measures. Secondly, we have presented a technique for modeling queue dynamics using signal phases as exogenous input to test suitability of a new set of phase-splits under abnormal conditions. Experimental results for a traffic accident show promise for this method, in that extended green-time at the affected leg results in less severe queue lengths. Further, by including queues from each leg in the state vector, the effect of the modified splits can be examined on all approaches simultaneously. Future work includes verifying an appropriate green extension to mitigate congestion under the constraint that other legs are not adversely affected by the modified split.

References

[1] Y. Kamarianakis, W. Shen, and L. Wynter, "Realtime road traffic forecasting using regime switching space time models and adaptive lasso," *Applied Stochastic Models in Business and Industry*, vol. 28, no. 4, pp. 297–315.

[2] M. Budišić, R. M. Mohr, and I. Mezić, "Applied Koopmanism," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 22, no. 4, 2012.

[3] C. W. Rowley, I. Mezic, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," *Journal of Fluid Mechanics*, vol. 641, pp. 115–127, 2009.

[4] B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.

[5] P. Rapp, T. Schmah, and A. Mees, "Models of knowing and the investigation of dynamical systems," *Physica D: Nonlinear Phenomena*, vol. 132, no. 1, pp. 133 – 149, 1999.

[6] C. Jones, *Whither Applied Nonlinear Dynamics?* Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 631–645.

[7] Y. Susuki, I. Mezic, F. Raak, and T. Hikihara, "Applied Koopman operator theory for power systems technology," *Nonlinear Theory and Its Applications, IEICE*, vol. 7, no. 4, pp. 430–459, 2016.

[8] J. Mann and J. N. Kutz, "Dynamic mode decomposition for financial trading strategies," *Quantitative Finance*, vol. 16, no. 11, pp. 1643–1655, 2016.

[9] J. L. Proctor and P. A. Eckhoff, "Discovering dynamic patterns from infectious disease data using dynamic mode decomposition," *International Health*, vol. 7, no. 2, pp. 139–145, 03 2015.

[10] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, 2010.

[11] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and N. J. Kutz, "On dynamic mode decomposition: theory and applications," *Journal of Computational Dynamics*, vol. 1, no. 2, pp. 391–421, 2014.

[12] Y. Susuki and I. Mezić, "Nonlinear Koopman modes and power system stability assessment without models," *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 899–907, 03 2014.

[13] F. Raak, Y. Susuki, I. Mezić, and T. Hikihara, "On Koopman and dynamic mode decompositions for application to dynamic data with low spatial dimension," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 12 2016, pp. 6485–6491.

[14] N. J. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. Society for Industrial and Applied Mathematics (SIAM), 2016.

[15] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic mode decomposition with control," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 1, pp. 142–161, 2016.

[16] Z. Amini, R. Pedarsani, A. Skabardonis, and P. Varaiya, "Queue-length estimation using real-time traffic data," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 11 2016, pp. 1476–1481.

[17] "Maryland open data portal," Maryland.gov, 08 2017. [Online]. Available: https://data.maryland.gov/browse?tags=crash