Koopman Operator Approach for Instability Detection and Mitigation

IEEE International Conference on Intelligent Transportation Systems

Esther Ling*, Lillian Ratliff**, Samuel Coogan*

* Georgia Institute of Technology
** University of Washington

The Traffic Control Loop

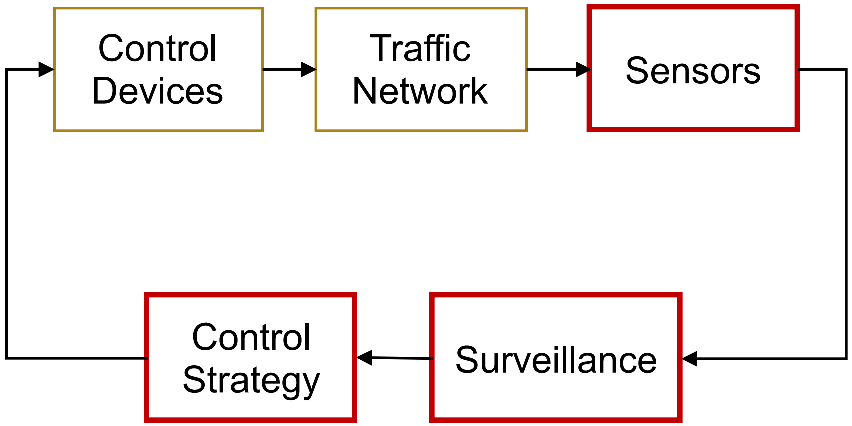
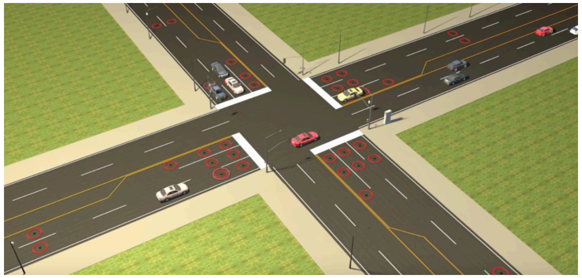


Image source: Sensys Networks



- Can we automate detection of imminent traffic congestion?
- Can we make data-driven models to "predict" effect of the control strategy?

Outline

- **Koopman Operator review**
- Two Applications:
  - Early detection of congestion
  - Capturing effect of signal timings in queue model

Koopman Operator

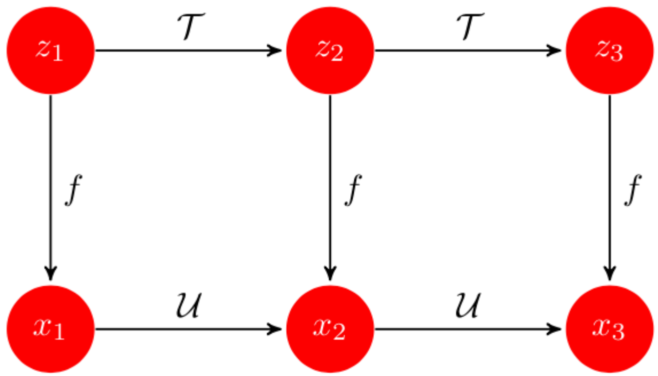
- Given a nonlinear discrete-time system

$$z_{k+1} = g(z_k)$$
$$x_k = f(z_k)$$

- Koopman Operator $\mathcal{U}$
  - Linear
  - Infinite-dimensional

$$f(z_{k+1}) = \mathcal{U} f(z_k)$$

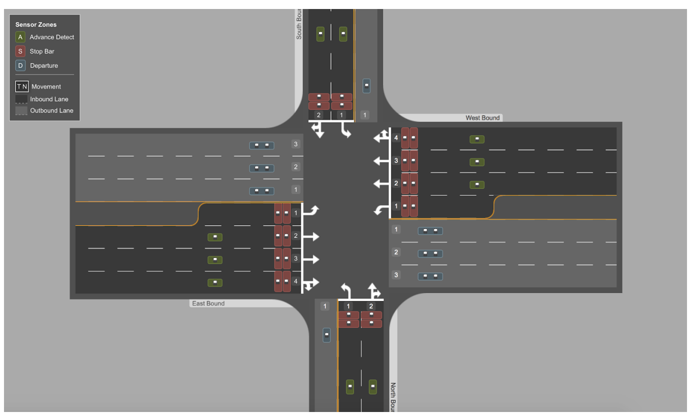
Evolution of States



Evolution of Functions on States (Observables)

Approximating an Infinite-Dimensional Operator using Data

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

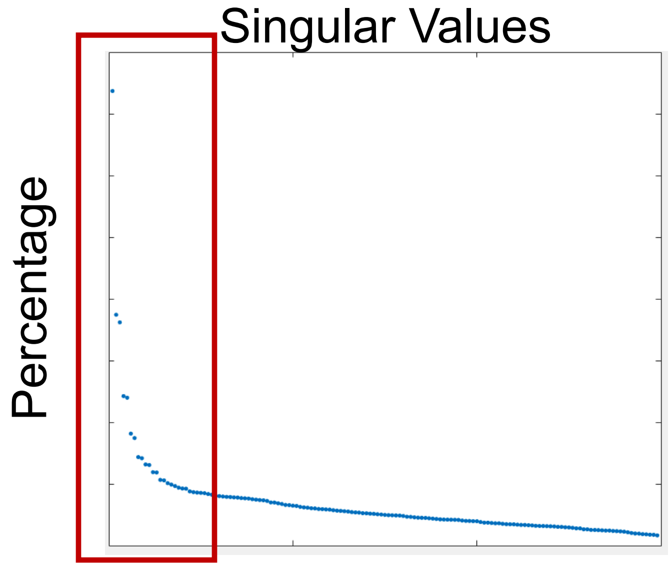


Suppose the sensor measurements are realizations of the observables

Dynamic Mode Decomposition

DMD: “approximate $\mathcal{U}$ using proxy matrix A by learning a locally-linear model”

$$X_2 = AX_1$$
$$A = X_2 X_1^\dagger$$
$$= X_2 V \Sigma^{-1} U^T$$



Abrupt decay in singular values

If A is large, high compute cost to perform eigen-decomposition:

- Use rank truncation in SVD ($\tilde{r} \leq r$)
- Use projection $\tilde{A} = U^T A U$

Koopman Operator Applications

Instability Analysis

- Eigenvalues

$$|\lambda| > 1$$

Indicates unstable dynamics

Spatio-temporal Information

- Modes

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

Provides relative spatio-temporal information

Prediction

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

Learn dynamics to predict future traffic

The Traffic Control Loop

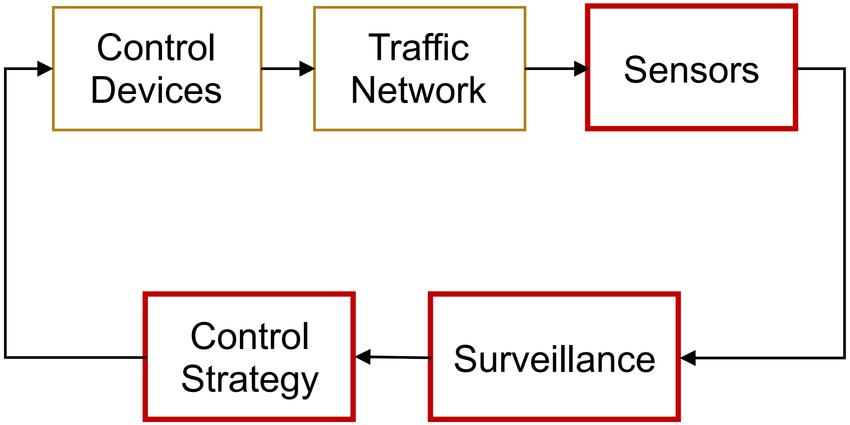
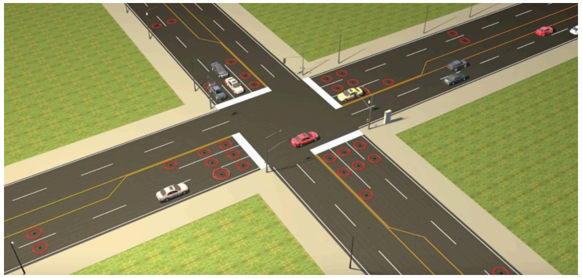


Image source: Sensys Networks



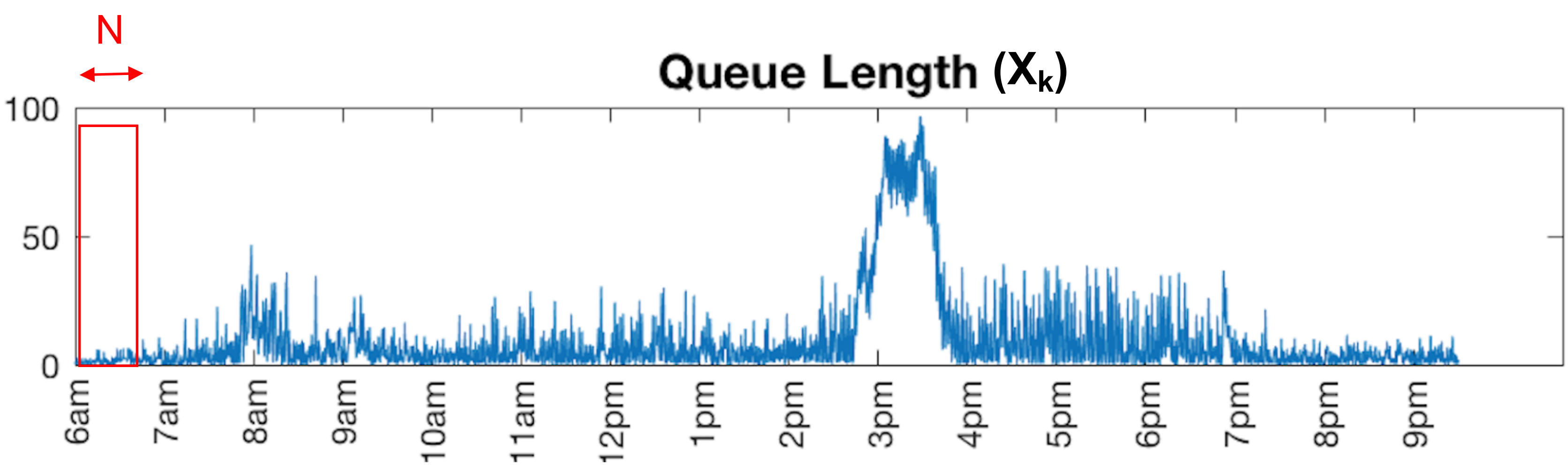
- Can we automate detection of imminent traffic congestion?
- Can we make data-driven models to “predict” effect of the control strategy?

Outline

- Koopman Operator review
- **Two Applications:**
  - **Early detection of congestion**
  - Capturing effect of signal timings in queue model

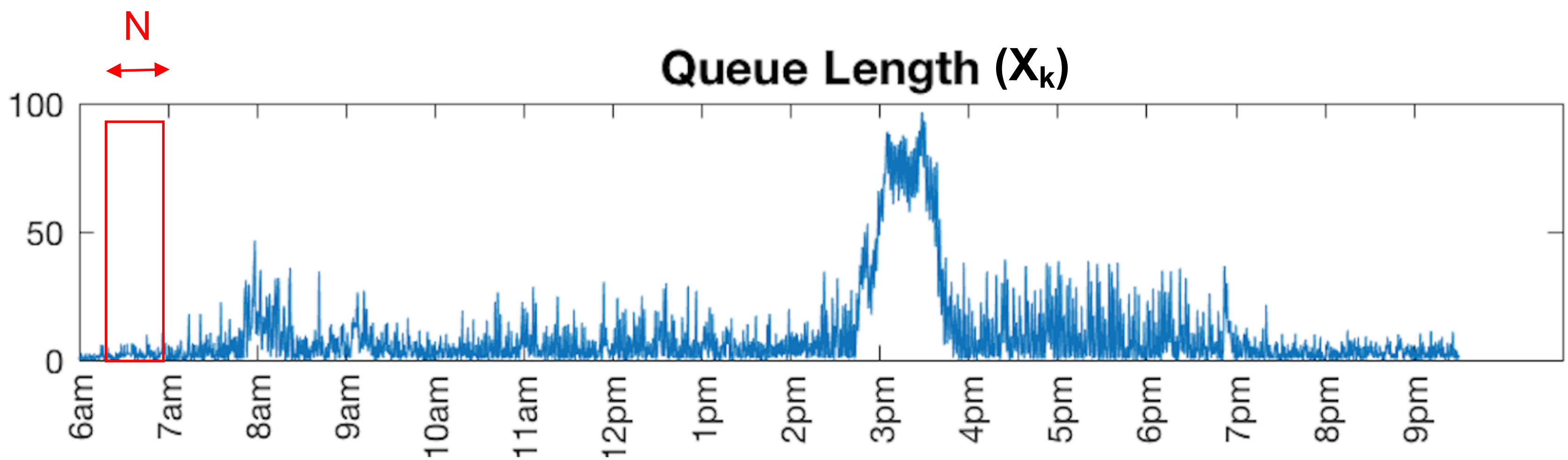
Instability Detection

- Local instability analysis to detect congestion
- How local? Specify the range of data to include, N
- Learn dynamics (A) in a rolling window
- Keep track of consecutive unstable eigenvalues



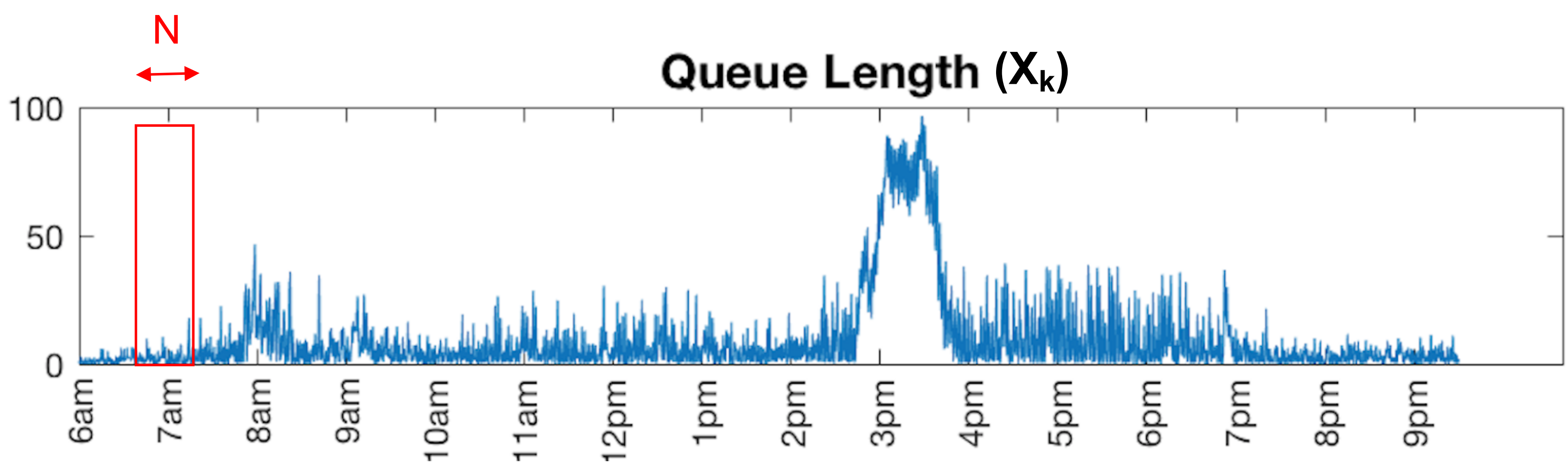
Instability Detection

- Local instability analysis to detect congestion
- How local? Specify the range of data to include, N
- Learn dynamics (A) in a rolling window
- Keep track of consecutive unstable eigenvalues



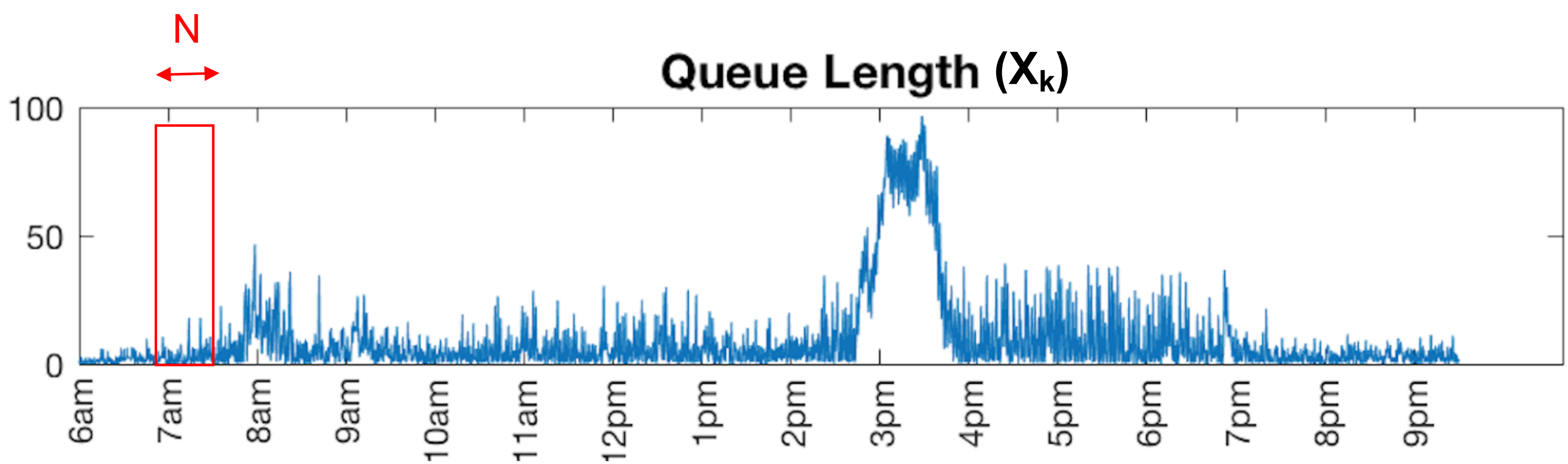
Instability Detection

- Local instability analysis to detect congestion
- How local? Specify the range of data to include, N
- Learn dynamics (A) in a rolling window
- Keep track of consecutive unstable eigenvalues



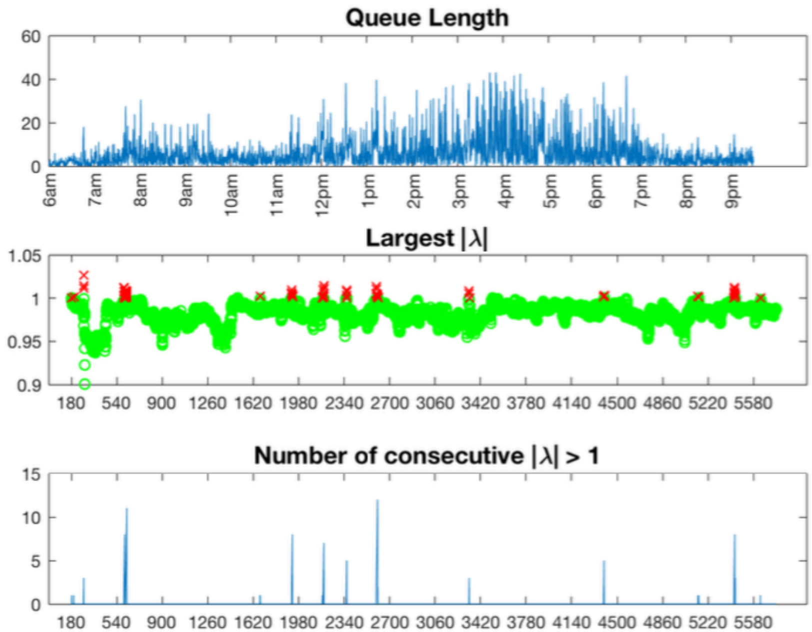
Instability Detection

- Local instability analysis to detect congestion
- How local? Specify the range of data to include, N
- Learn dynamics (A) in a rolling window
- Keep track of consecutive unstable eigenvalues

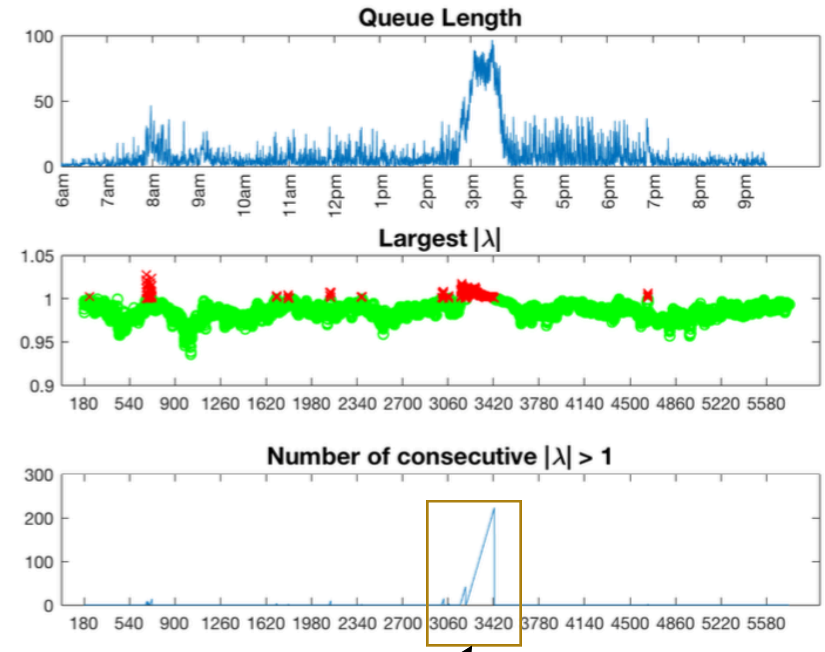


Instability Detection

Normal Day



Accident Day



Indicates growth for 33 minutes
(10s step between rolling window)

Outline

- Koopman Operator review
- **Two Applications:**
  - Early detection of congestion
  - **Capturing effect of signal timings in queue model**

Effect of signal timings in queue model

- Notice that the queue starts to clear at 3.30pm
- Scheduled change in timing plan at 3.30pm
- Did the extended green time for congested leg play a role?

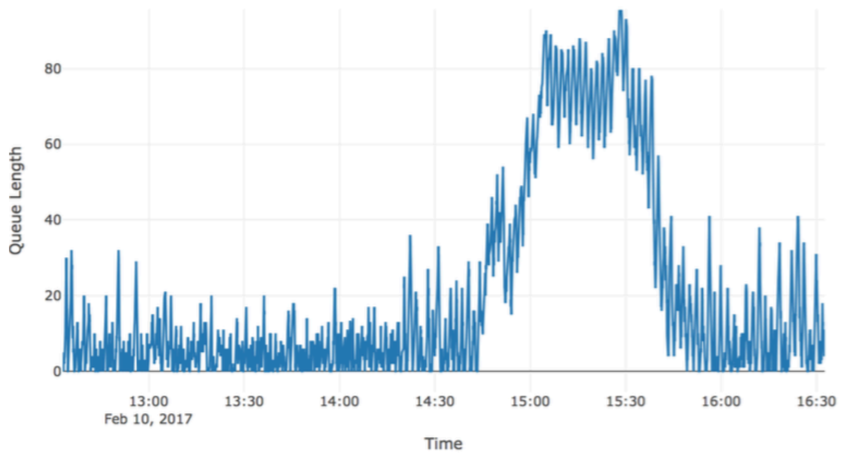


TABLE I
TIMING PLAN FOR THE INTERSECTION. (1-EBLT, 2-EB, 4-SB, 5-WBLT, 6-WB, 8-NB). (LEFT) 9.30AM-3.30PM, 110S. (RIGHT) 3.30PM-7.00PM, 120S. PHASES 1, 2, 5 AND 6 ARE IN BARRIER 1, WHILE PHASES 4 AND 8 ARE IN BARRIER 2.

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

Effect of signal timings in queue model

Learn A and B using original x_k and u_k

- **Do A and B learn a good model?** Reconstruct $\{x_2, \ldots, x_N\}$ using initial condition x_1 and $\{u_1, \ldots, u_N\}$
- **What is the effect of a modified phase-split?** Reconstruct $\{x_2, \ldots, x_N\}$ using initial condition x_1 and modified $\{u_1, \ldots, u_N\}$

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

$$x_k \in \mathbb{R}^4$$

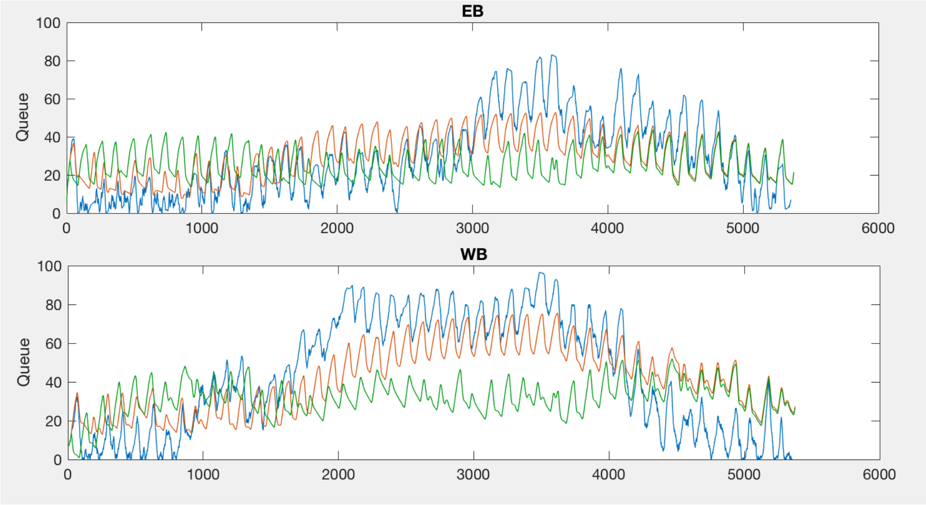
$$u_k = \begin{bmatrix} u_k(1) \\ u_k(2) \\ \vdots \\ u_k(12) \end{bmatrix}$$

where

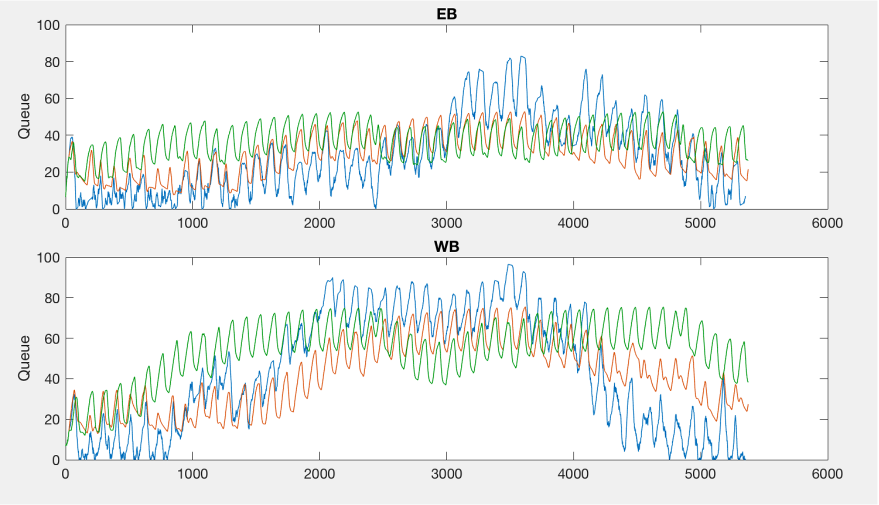
$$u_k(i) = \begin{cases} 0, & \text{if } u_k(i) = \text{red} \\ 1, & \text{if } u_k(i) = \text{green or yellow} \end{cases}$$

Effect of signal timings in queue model

- Longer green times for congested leg ⇒ faster queue mitigation
- Can visualize effect on all 4 legs with one model



Queue Plots for Congested Legs
(longer green time)



Queue Plots for Congested Legs
(shorter green time)

Legend

Blue = Original queues

Red = Reconstructed queues using original signal phases

Green = Reconstructed queues using modified signal phases

Summary and Q&A

- Koopman Operator framework for data-driven modeling
- Applications:
  - Automated early detection of traffic congestion
  - Modeling queue dynamics with signal phases to anticipate effect of modified phase-splits
- Future Directions:
  - What is an adequate amount of green time extension?
  - Model is currently intersection-level. Can this be extended to include a network-level model?