G-Active

Estimation and prediction of traffic dynamics with data-driven low-dimensional model based on SUMO

Dr Jakub Krol (UCL), Dr Bani Anvari (UCL) and Prof Roberto Lot (UoS)

SUMO User Conference 2019 – New Tools Session, 15 May 2019

EPSRC Engineering and Physical Sciences Research Council

University of Southampton

Imperial College London

UCL

Structure

- Motivation
- Proposed approach
- Estimation and prediction results
- Conclusions and further work

Motivation

Motivation – G-Active project

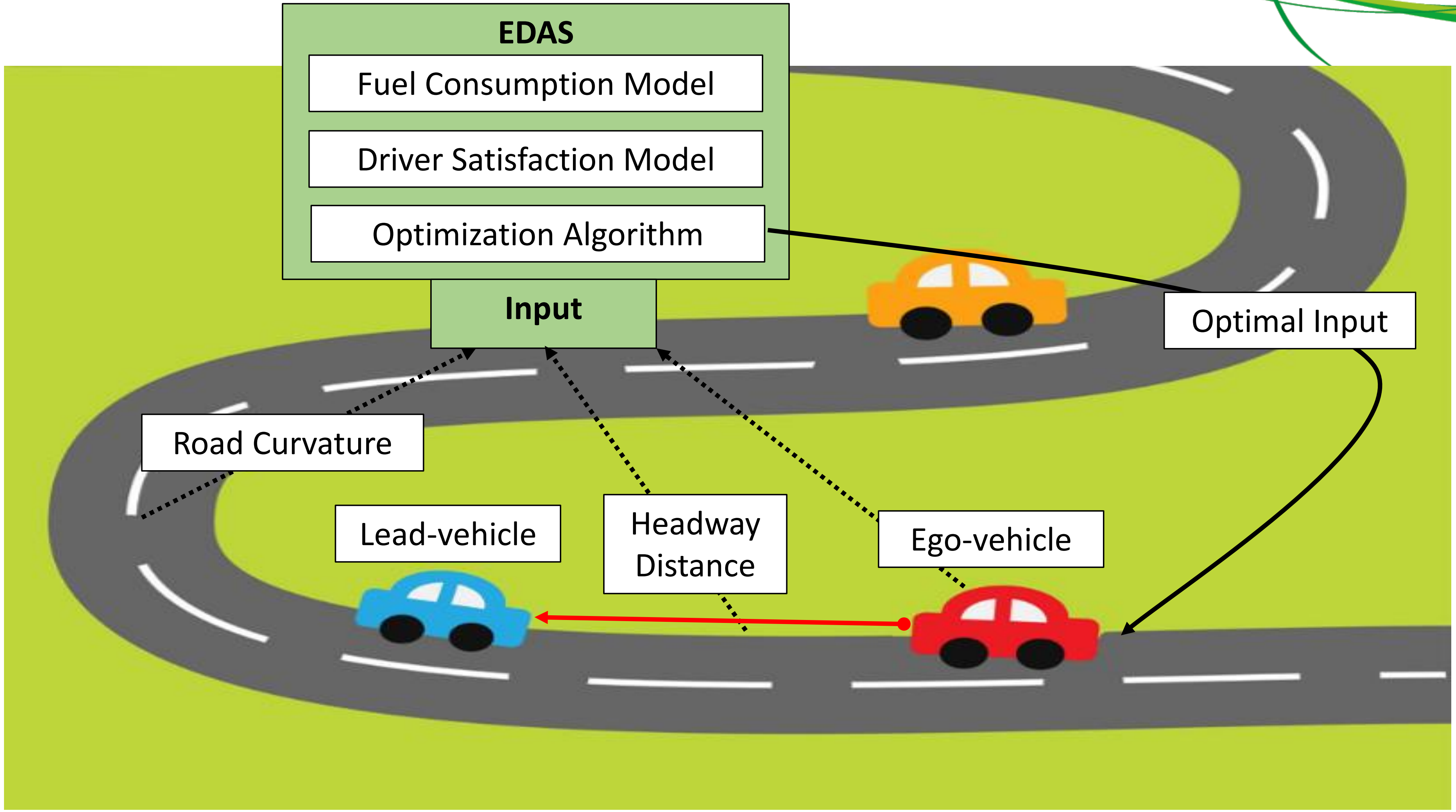
Issues

- Research has estimated that 5-10% of fuel can be saved if drivers do eco-driving
- The state-of-the-art eco-driving techniques do not take into account the naturalistic behaviour of human drivers

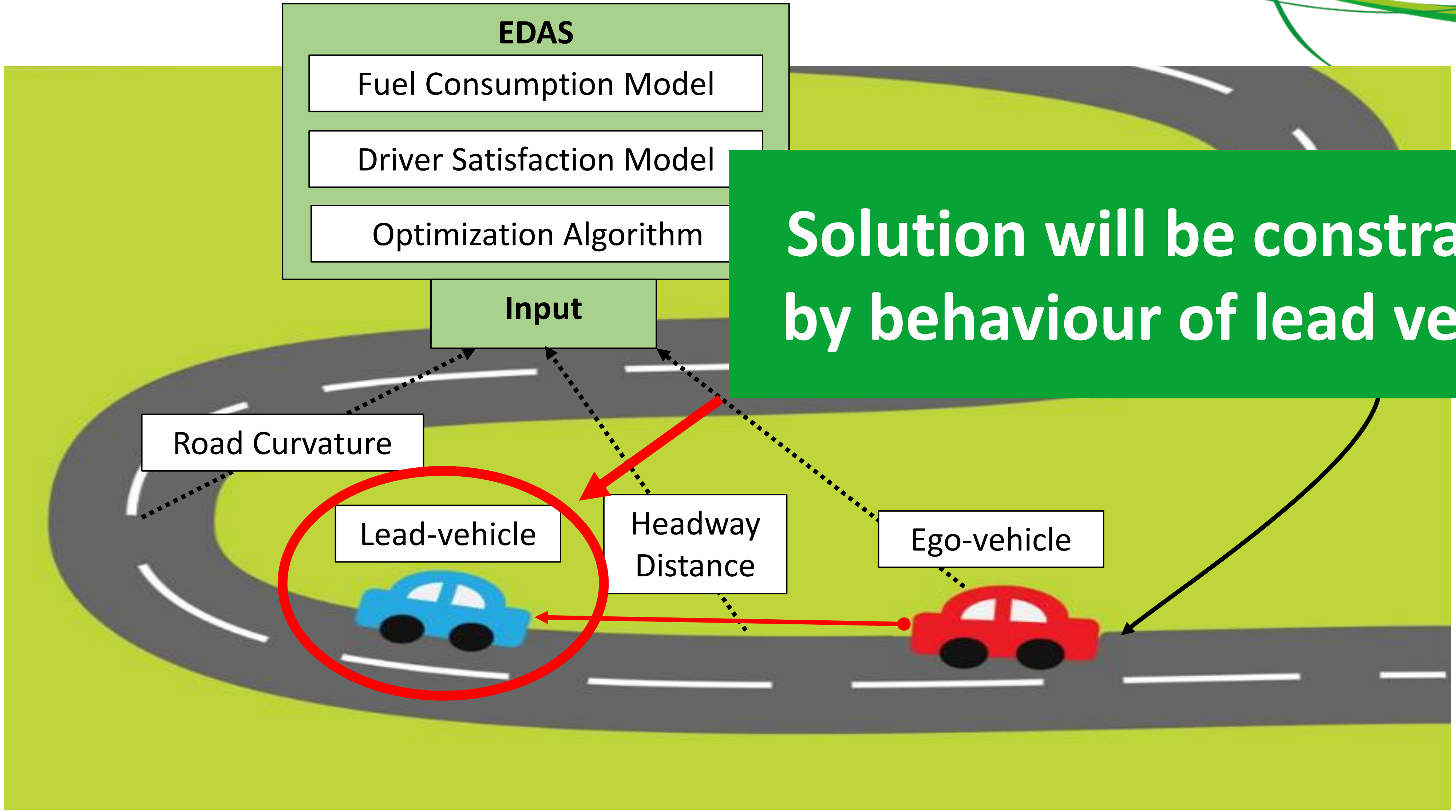
Our Solution

- A unified driver model is proposed which describes the driver preference during car following and cornering cases
- Speed optimization problem is formulated considering both fuel economy and driver preference

Motivation


EDAS
Fuel Consumption Model
Driver Satisfaction Model
Optimization Algorithm
Input
Optimal Input
Road Curvature
Lead-vehicle
Headway Distance
Ego-vehicle

Motivation



Solution will be constrained by behaviour of lead vehicle

Motivation

Prediction framework

1. Need to be able to make predictions based on available live measurements such as those from induction loops

2. We want to be able to sample any location in the network and obtain estimation and prediction corresponding to that point

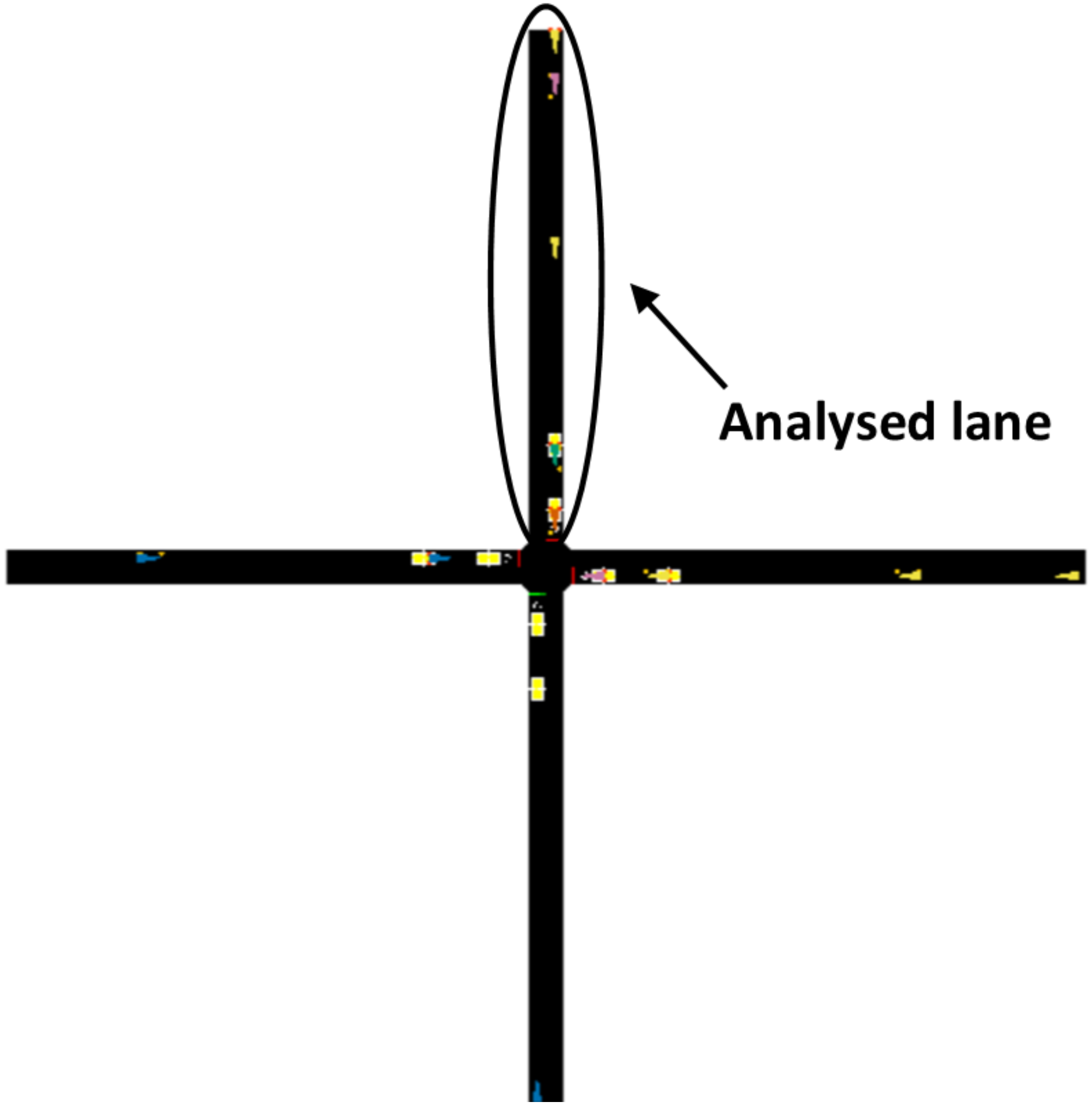
3. Need to be able to obtain estimate and prediction fast

Proposed Approach

Test scenario

Vehicles approaching a junction

- Simple scenario to facilitate understanding and validity of the methodology
- Dimensions: 95.25m long, 3.20m wide
- Induction loop at 89.25m (only measurement available to investigate traffic state)
- Vehicles enter lane on average every 10s, leading to congested traffic conditions
- Vehicles follow Intelligent Driver Model (IDM)
- Simulation time is 2h with sampling rate of 0.1s resulting in 72,000 measurements
- **Train data**: t > 1h, **Test data**: t < 1h



Methodology

Prediction framework

1. Need to be able to make predictions based on available live measurements such as those from induction loops
2. We want to be able to sample any location in the network and obtain estimation and prediction corresponding to that point
3. Need to be able to obtain estimate and prediction fast

Model-Based Estimation

Assume that investigated system can be expressed as

$$a_{t+1} = A(a_t) + v_t$$
$$y_t = C(a_t) + w_t$$

Given:

- Imperfect dynamic model
- Noisy measurements

What is the optimal estimate of the state?

Solve:

$$\min_a \sum_t (||a_{t+1} - A(a_t)||_2 + ||y_t - C(a_t)||_2)$$

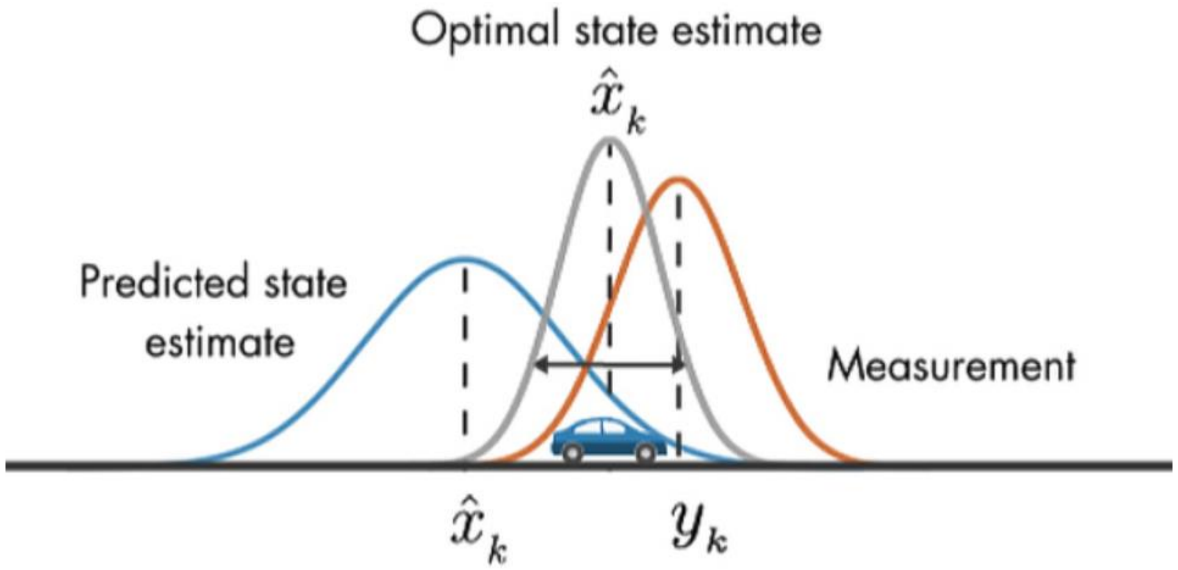
Kalman Filter

Kalman filter gives the optimal state estimate under the following assumptions:

- **Linear model** and **Gaussian distribution** of inaccuracies

Two-step procedure at each time step:

- Model **prediction**
- Measurement **update**



Source: *Ulusoy, M. Understanding Kalman Filters, Part 1: Why Use Kalman Filters?*

Model-Based Estimation

Assume that investigated system can be expressed as

$$a_{t+1} = A(a_t) + v_t$$
$$y_t = C(a_t) + w_t$$

Given:

- Imperfect dynamic model
- Noisy measurements

What is the optimal estimate of

Solve:

$$\min_a \sum_t (||a_{t+1} - A(a_t)||_2 + ||y_t - C(a_t)||_2)$$

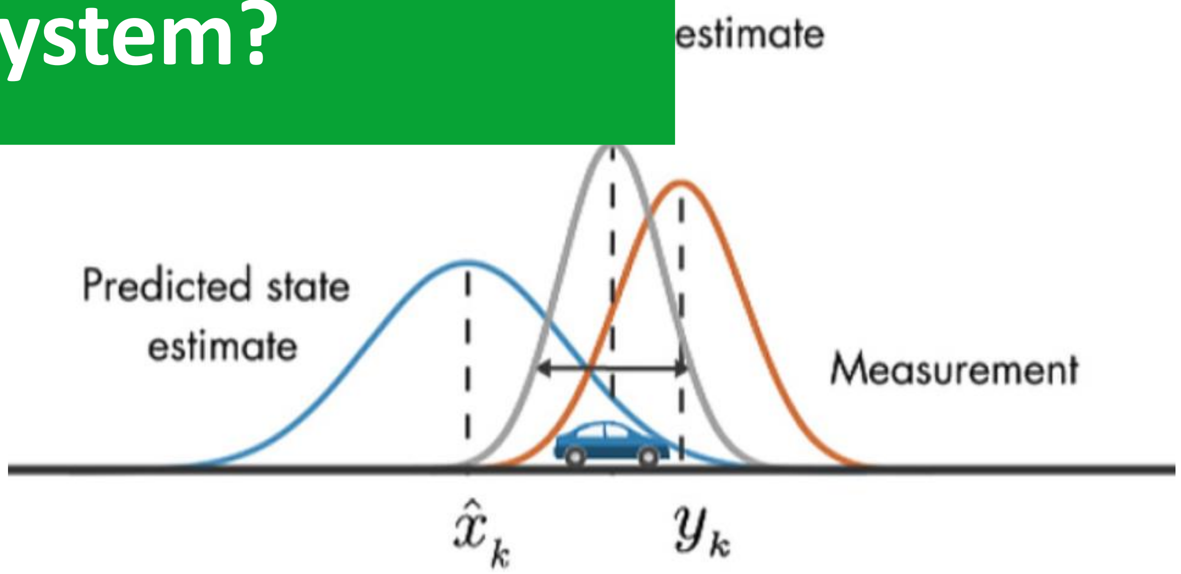
Kalman Filter

Kalman filter gives the optimal state estimate under the following assumptions:

- **Linear model** and **Gaussian distribution** of inaccuracies

Two-step procedure at each timestep:

What should be the state of the system?



Source: *Ulusoy, M. Understanding Kalman Filters, Part 1: Why Use Kalman Filters?*

Methodology

Prediction framework

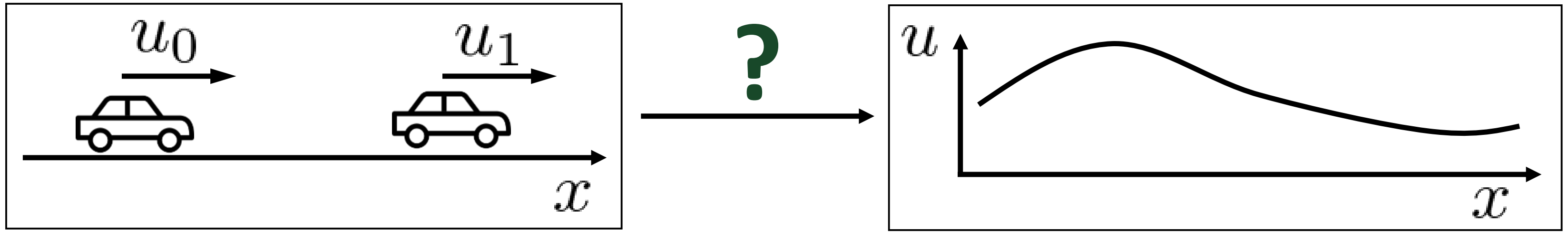
1. Need to be able to make predictions based on available live measurements such as those from induction loops

2. We want to be able to sample any location in the network and obtain estimation and prediction corresponding to that point

3. Need to be able to obtain estimate and prediction fast

Constructing continuous velocities

Problem:



Adaptive Smoothing Method (Treiber et al, 2002):

- Spatiotemporal domain perturbations propagate downstream at free flow velocity and upstream at congested velocity:

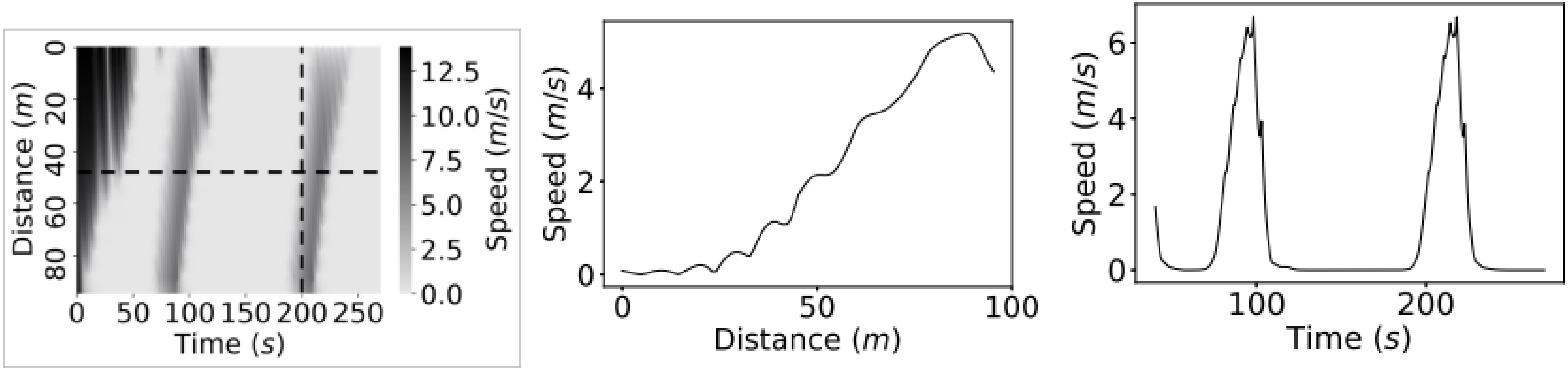
$$z_{free} = \frac{1}{\sum \Phi} \sum_{i}^{n} \sum_{j=j_{min}}^{j_{max}} \Phi(x_i - x, t_j - t - x/c_{free}) z_{i,j}$$

  Use the interpolated quantity: $z_{interp} = w z_{free} + (1-w) z_{cong}$

- Exponential kernel used to decay points which are far away. We replace it with: $\Phi(x,t) = \dfrac{1}{x^2 + t^2}$

Ref.: Treiber, Martin, and Helbing, Dirk. "An adaptive smoothing method for traffic state identification from incomplete information." Interface and Transport Dynamics. Springer, Berlin, Heidelberg, 2003. 343-360.

Constructing continuous velocities



(a) (b) (c)

Figure: (a) Spatio-temporal diagram obtained from the modified Adaptive Smoothing Method;
(b) spatial snapshot at t = 200s;
(c) temporal snapshot at x = 48m.

Note that locations of the snapshots are represented on the spatio-temporal diagram.

Methodology

Prediction framework

1. Need to be able to make predictions based on available live measurements such as those from induction loops

2. We want to be able to sample any location in the network and obtain estimation and prediction corresponding to that point

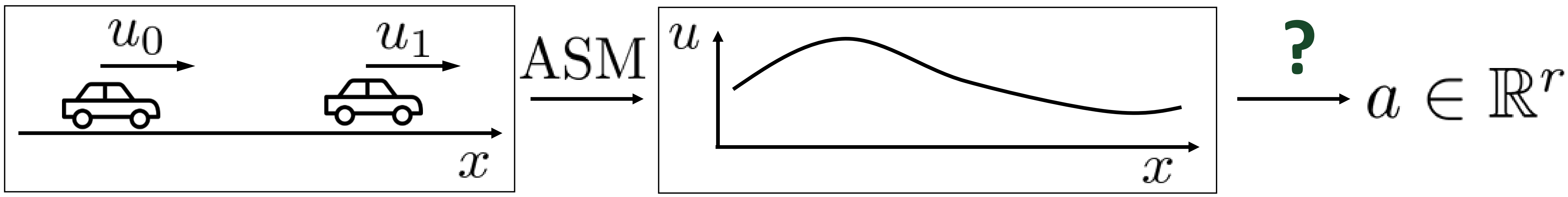
3. Need to be able to obtain estimate and prediction fast

Dimensional reduction

- If the state space consists of the discretised spatial domain, the dimension of the problem will grow large at the network level
- Instead we express velocity field as sum of time-invariant structures

$$u(x,t) \approx \sum_{i=1}^{r} a_i(t)\Phi_i(x)$$

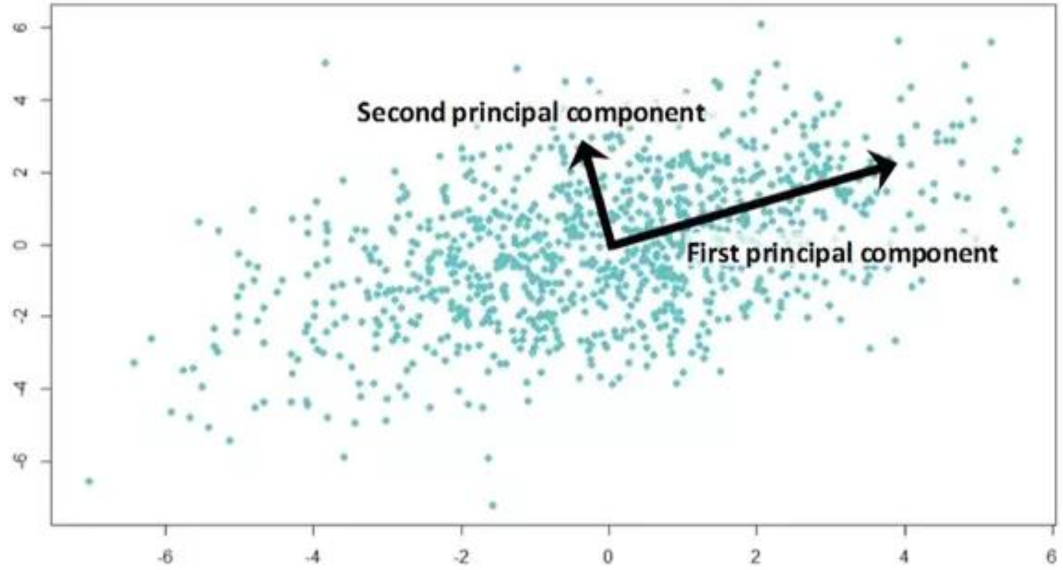
- State space will consist of a_i where r can be chosen freely



Dimensional reduction

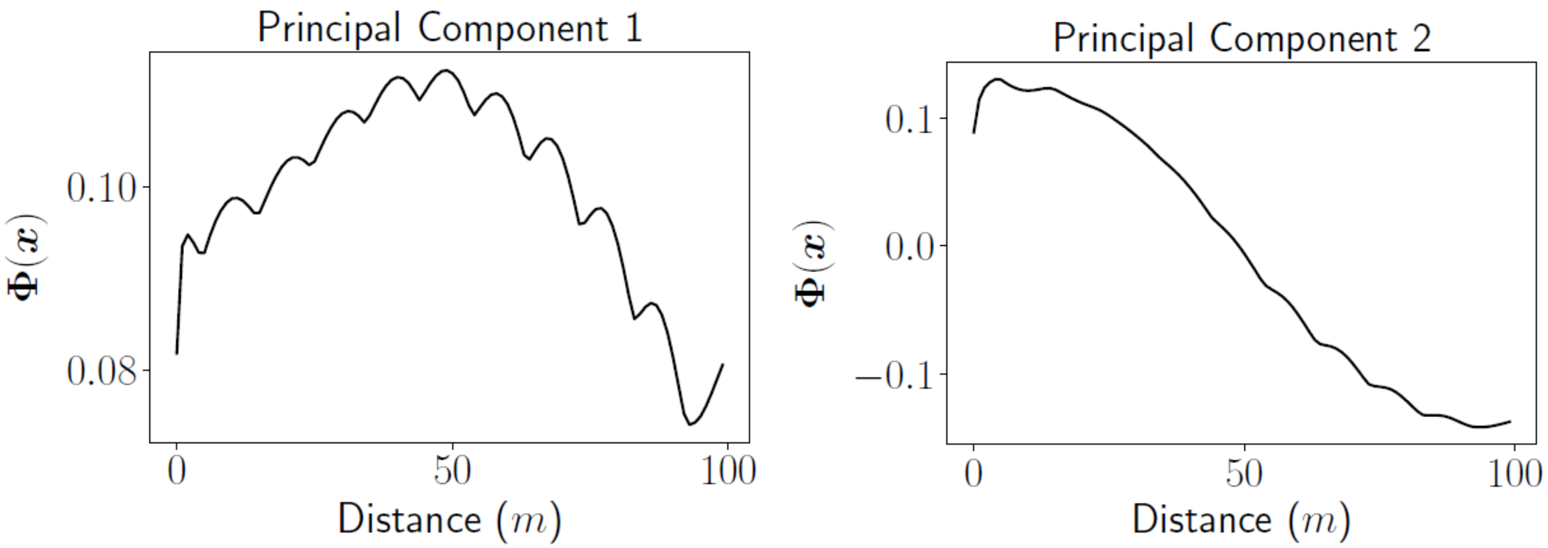
Principal Component Analysis

- Principal Component Analysis gives structures Φ_i and coefficients a_i which are optimal with respect to l^2 – norm

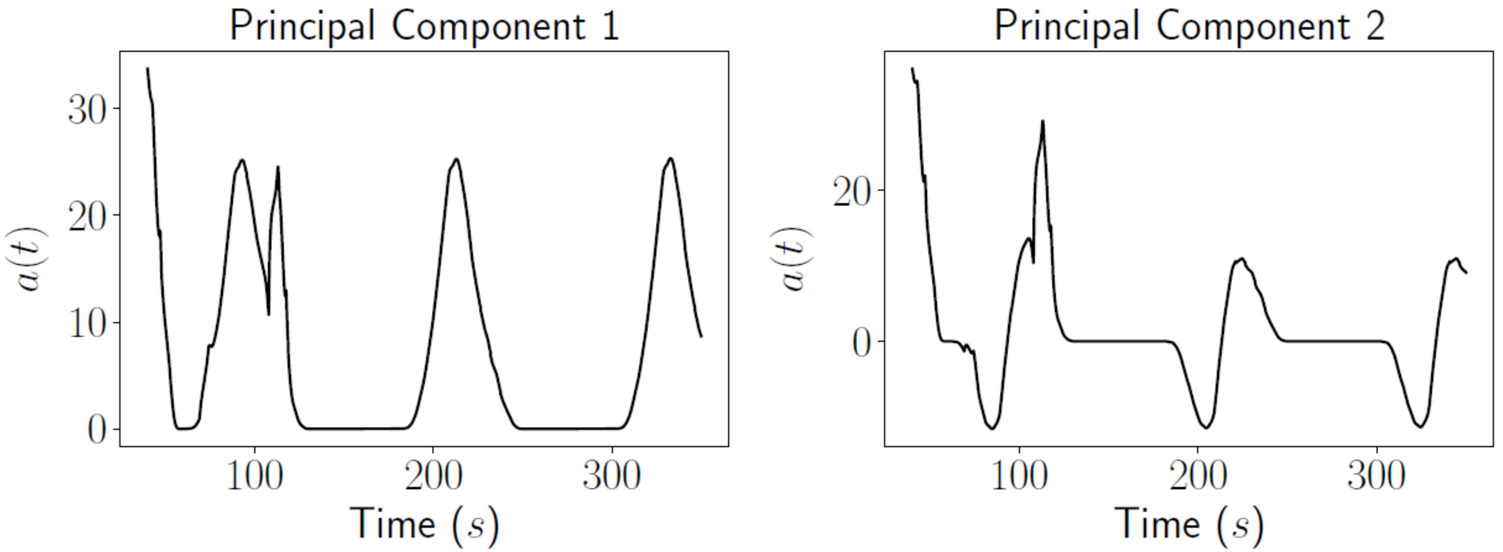


Source: *Nazrul, Syed S. The DOs and DON'Ts of Principal Component Analysis*

Spatial Components

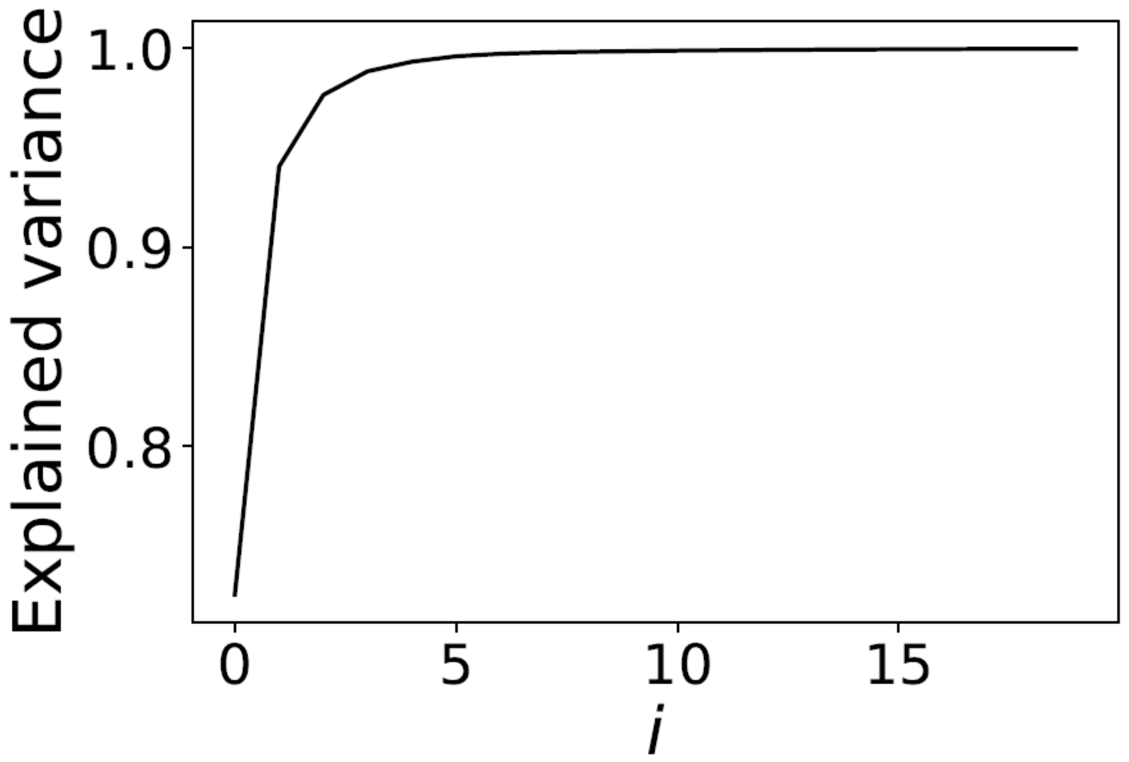


Temporal Coefficients



Dimensional reduction and linear model

- Number of principal components are chosen based on explained variance, where six components explain 99% of variance



- Computation time will be reduced by utilising the linearity of the Kalman filter
- Get models by solving following on train data:

$$\min_{A} \sum_{t} ||a_{t+1} - Aa_t||_2$$

$$\min_{C} \sum_{t} ||\boldsymbol{y}_t - Cx_t||_2$$

where $\boldsymbol{y}_t = [y_t, y_{t-1}, \ldots, y_{t-5}]^\top$

- Furthermore we provide noise covariance matrices

$$\boldsymbol{Q} = \boldsymbol{e}_p \boldsymbol{e}_p^\top \qquad \boldsymbol{R} = \boldsymbol{e}_m \boldsymbol{e}_m^\top$$

where

$$\boldsymbol{e}_p = \begin{bmatrix} \boldsymbol{a}_1 - \boldsymbol{A}\boldsymbol{a}_0 & \ldots & \boldsymbol{a}_N - \boldsymbol{A}\boldsymbol{a}_{N-1} \end{bmatrix}$$

$$\boldsymbol{e}_m = \begin{bmatrix} \boldsymbol{y}_0 - \boldsymbol{C}\boldsymbol{a}_0 & \ldots & \boldsymbol{y}_N - \boldsymbol{C}\boldsymbol{a}_N \end{bmatrix}$$

Methodology

Framework training

- Run **SUMO simulation** to obtain underlying discrete data
- Construct **continuous velocity field**
- Calculate **principal components**
- **Project** high-dimensional data to obtain **low-dimensional state**
- Find **linear dynamics** matrices and corresponding error covariance matrices

Application

Estimation:

- At each time step apply **Kalman filter** to get state estimate

Prediction:

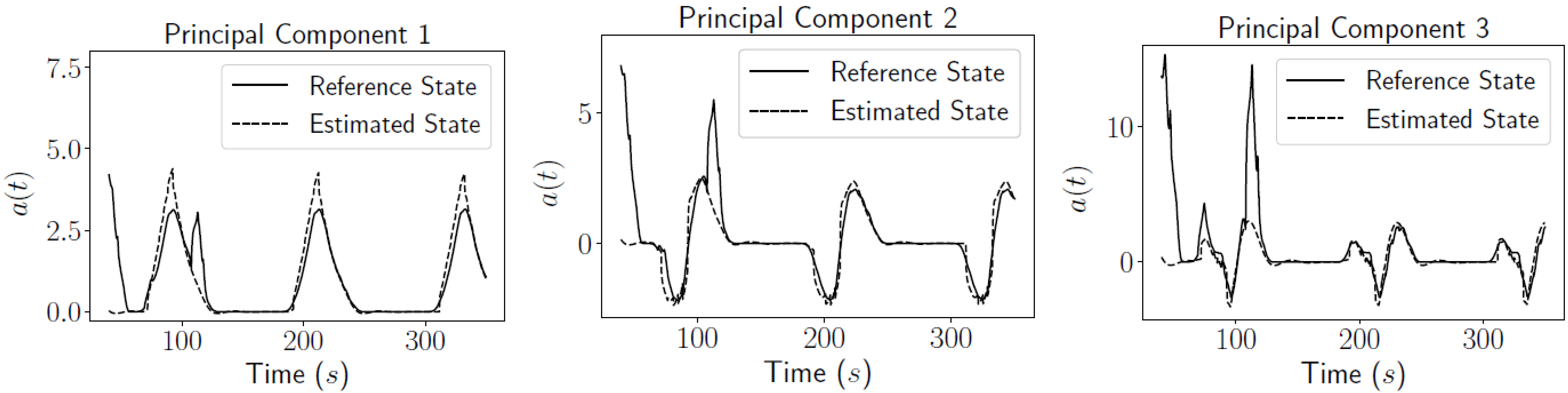
- Use linear model to **propagate state forward in time**
- **Recompose** the high dimensional **velocity field** using predicted state
- **Integrate along spatiotemporal diagram** from initial position

Estimation and Prediction Results

Estimation

Kalman filter modification

- In the two step procedure, when measurements are not available update step is omitted



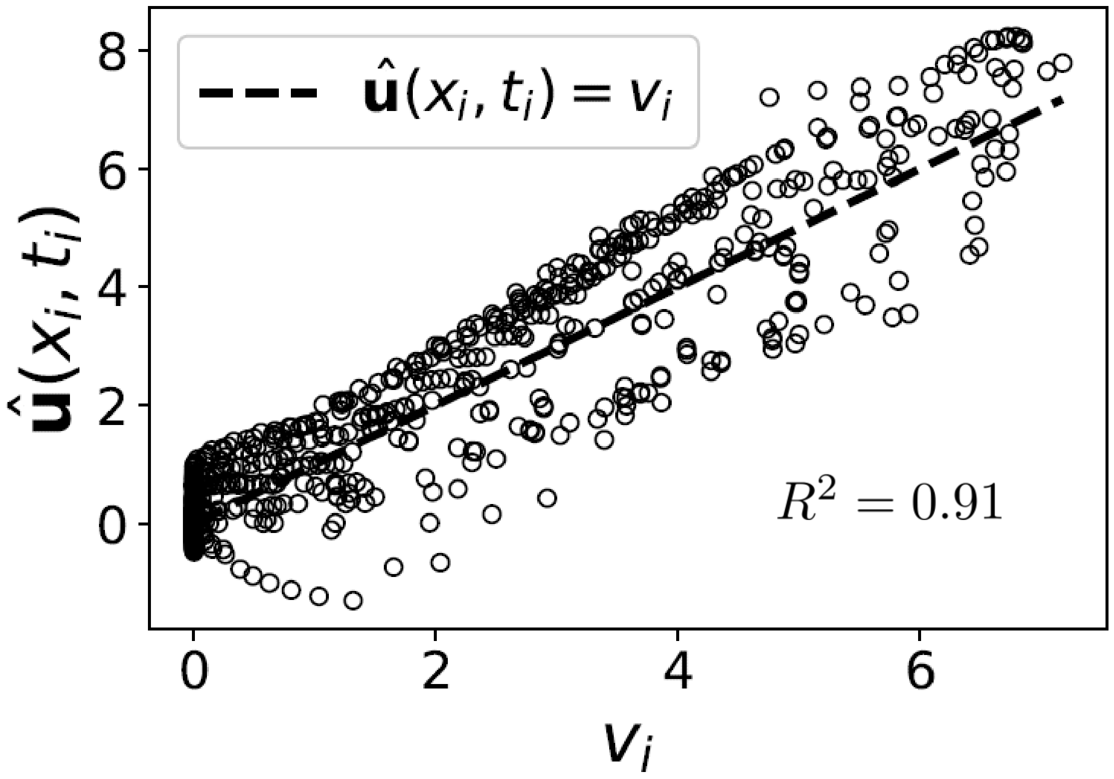
Relative error

$$\epsilon_i^2 = \frac{\sum_{t>150}(a_i(t) - \hat{a}_i(t))^2}{\sum_{t>150}(a_i(t))^2}$$

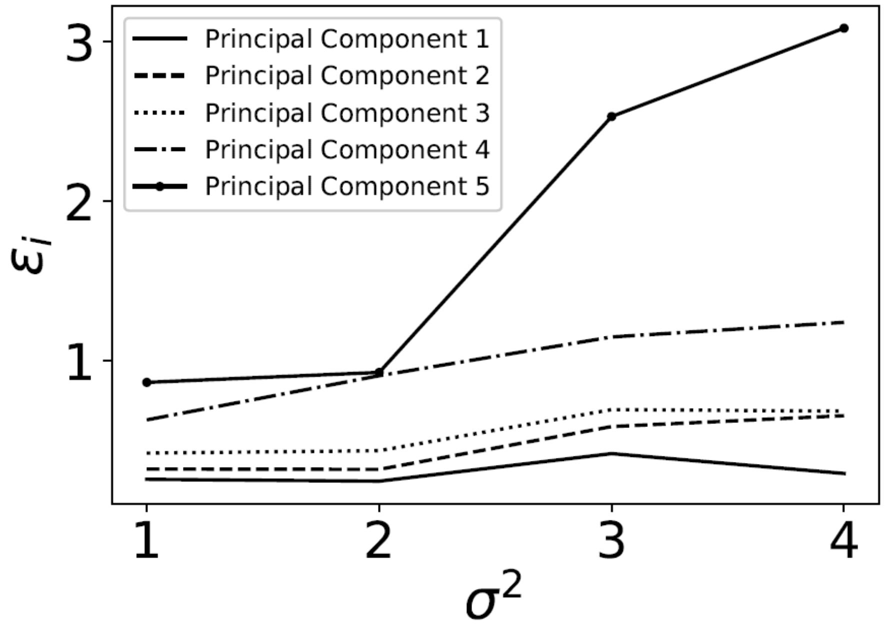
Principal Component	1	2	3	4	5	6
ϵ_i	0.242	0.284	0.382	0.584	0.856	0.789

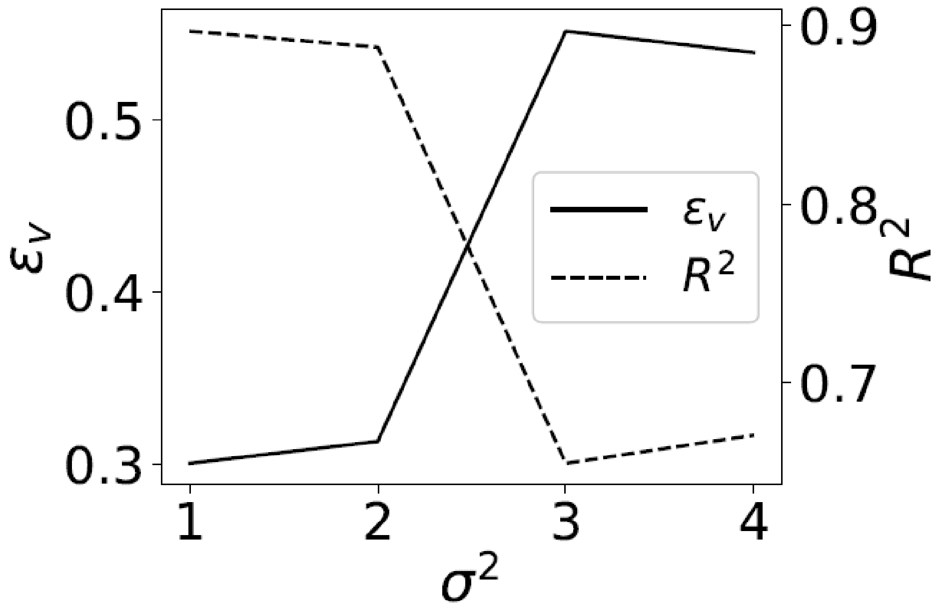
Estimation

Discrete metric – perfect drivers



Imperfect drivers



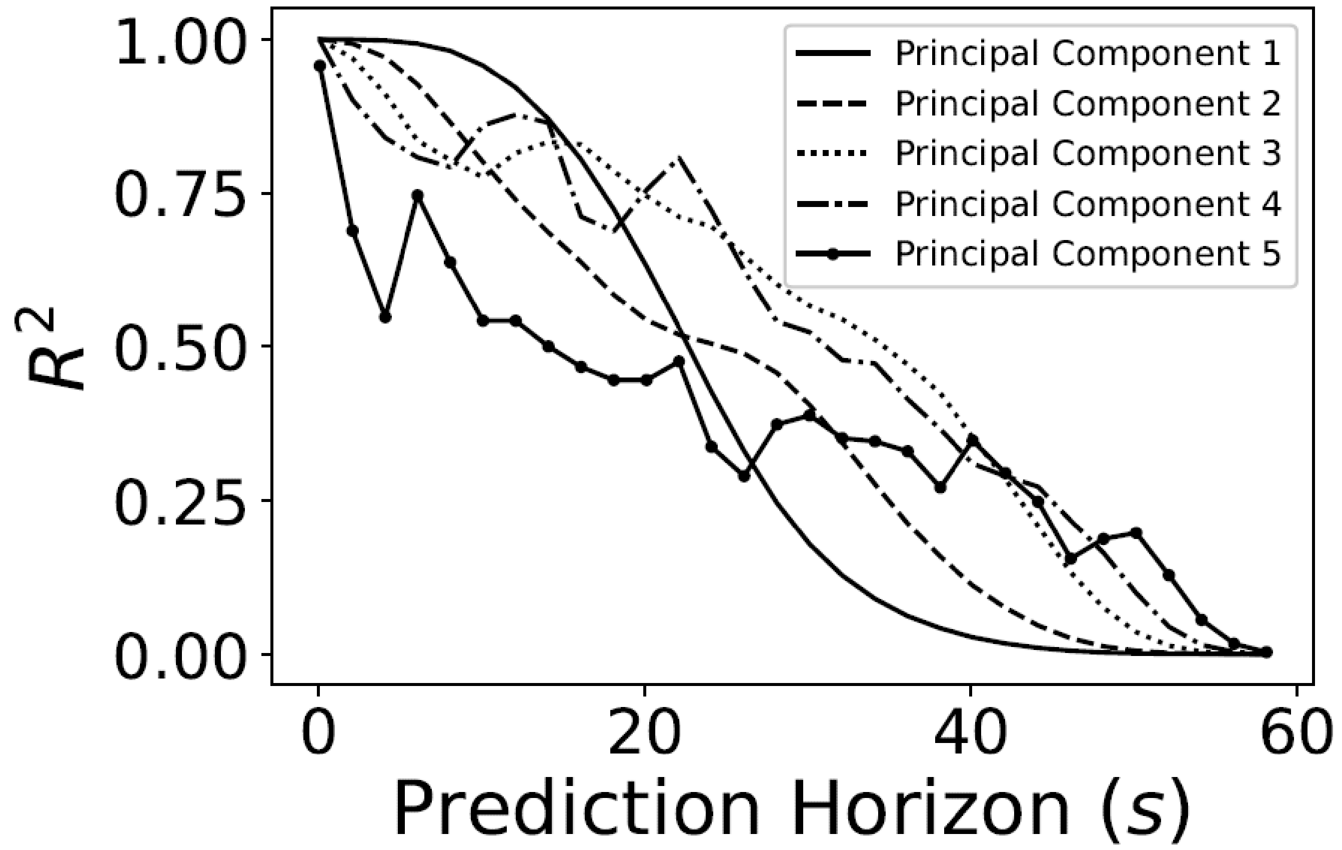


Imperfect drivers:

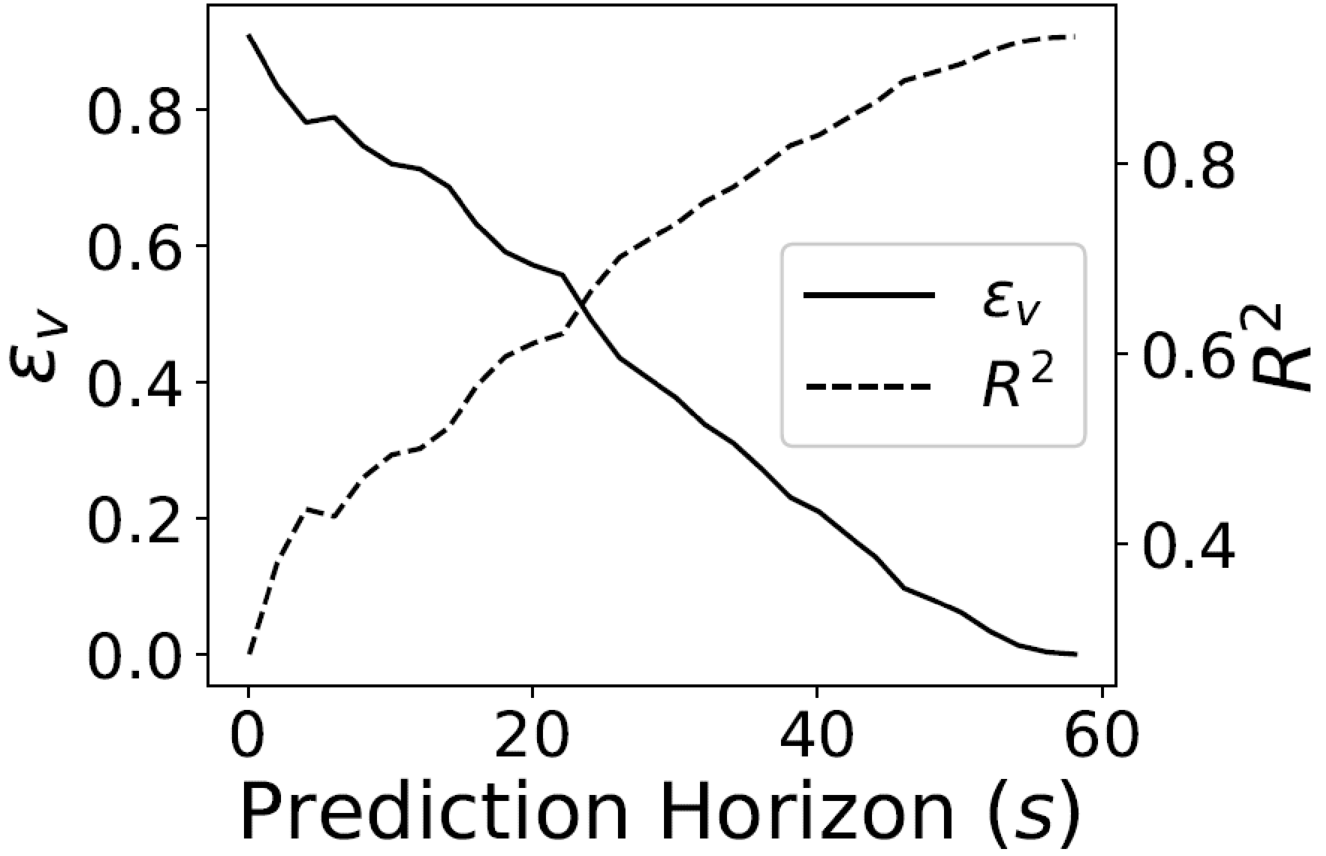
- For desired speed $u^*(t)$ given by IDM, the speed of each vehicle is modified such that $u(t) = ku^*(t)$ where $k \sim \mathcal{N}(1, \sigma^2)$

Prediction

Projection Coefficients



Discrete Metric



Conclusions and Future Work

Conclusions

- The methodology which estimates velocity profiles based on the real-time information from induction loops was presented.
- Implementation is linear and low-dimensional; suitable in situations where computational cost and simplicity are of high importance.
- Analysis on a problem of vehicles approaching traffic junction in a congested traffic showed that, due to semi-linear dynamics, such system can be well estimated with few structures and linear Kalman filter.
- Robustness in the estimation of key dynamical features when applied to dynamics with driver imperfections.

Future work

Simulated Urban Scenario

- Selly Oak, Birmingham
- Birmingham makes induction loop measurements publicly available, allowing us to test validity of estimation and prediction framework

Potential method improvements

- **Nonlinear Dynamics:** IDM-based Galerkin projection, Machine Learning
- **Alternative dimensional reduction techniques**
- **Improvements to ASM:** Smoothing parameters, Different kernel choice

Many thanks for your attention.

Dr Jakub Krol
@JakubKrol5
j.krol@ucl.ac.uk