The Thesis committee for Cesar N. Yahia certifies that this is the approved version of the following thesis:

Unmanned Aerial Vehicle Path Planning for Traffic Estimation and Detection of Non-Recurrent Congestion

APPROVED BY

SUPERVISING COMMITTEE:

_____

Stephen D. Boyles, Supervisor

_____

Christian G. Claudel

Unmanned Aerial Vehicle Path Planning for Traffic Estimation and Detection of Non-Recurrent Congestion

by

Cesar N. Yahia

Thesis

Presented to the Faculty of the Graduate School
of the University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science in Engineering

The University of Texas at Austin
August 2018

Acknowledgments

This thesis would not have been possible without the continuous guidance and support of my advisor Prof. Stephen D. Boyles. Thank you, Dr. Boyles, for being a great mentor and friend.

I would also like to thank Prof. Christian Claudel for helpful comments on this research effort and for all the tea-time entertainment in the 6.2 suite. In addition, I would like to thank Prof. Paola Passalacqua for her contagious passion for research and all the excitement she brings to our project meetings.

I appreciate the support of all my friends in Austin. Special thanks to Ahmad AlAmmouri and Natalia Zuniga-Garcia for constant emotional and mental support.

Srijith Balakrishnan was very kind to provide office space that was crucial to the proper completion of this thesis.

Venktesh Pandey helped with Chapter 4 of this thesis. Thank you!

I appreciate the administrative help of Lisa Macias and Velma Vela. They made sure I registered on time and guided me through the graduation process.

My parents and siblings have always been a source of love and inspiration. I would like to thank them for all the valuable life lessons.

I gratefully acknowledge the financial support of the Center for Advanced Multimodal Mobility Solutions and Education (CAMMSE), the Data-Supported Transportation Operations and Planning (DSTOP) center, the consortium of Cooperative Mobility for Competitive Megaregions (CM2), and the National Science Foundation under grant no. 1636154.

Unmanned Aerial Vehicle Path Planning for Traffic Estimation and Detection of Non-Recurrent Congestion

by

Cesar N. Yahia, M.S.E
The University of Texas at Austin, 2018
Supervisor: Stephen D. Boyles

Unmanned aerial vehicles (drones) can be used in traffic and road monitoring applications. We investigate the benefit of using drones for simultaneous traffic state estimation and incident detection. Specifically, we propose a coupled planning and estimation framework where we adaptively navigate a drone to minimize the uncertainty on parameter and traffic state estimates. We show that the use of a drone provides significant improvement in incident detection under congested conditions. Without a drone, the estimation procedure in congested conditions is not able to distinguish between observations due to congestion under normal operating conditions and similar observations due to a reduction in capacity.

To plan for multiple drones working in parallel, we investigate methods for partitioning a network such that each drone operates in a particular subnetwork. The partitioning objectives are to minimize inter-flow between partitions and to create balanced subnetworks. We implement a flow-weighted spectral partitioning algorithm and show that it performs better than simple agglomerative heuristics.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background & Motivation

Emerging technologies, such as unmanned aerial vehicles (drones), are changing the transportation landscape. A drone is an autonomous aircraft that can be equipped with cameras and communication devices. Recent advances in image processing techniques enable the use of drones for real-time traffic monitoring. In particular, researchers across the world are using drones to examine road conditions and to measure traffic flow characteristics such as speed and density [1].

Model-driven traffic estimation is the process of determining traffic state variables (density, speed, and flow) from partially observed noisy data and models that represent traffic dynamics [2]. These estimation methods use nonlinear filters to adjust traffic state model predictions based on observed sensor measurements. The adjustments rely on the relative uncertainty in model predictions compared to the noise on sensor measurements.

While estimating the traffic state using ground sensor data (e.g., loop detectors or GPS equipped probe vehicles) has been extensively studied, drones provide novel sensing capabilities that can improve the quality of resulting estimates. Specifically, drones can obtain targeted observations in locations where there is a high uncertainty on model predictions or sensor measurements. In other words, we are capable of routing the drone towards regions with a high variance on

the traffic state variables to generate accurate observations of the true underlying states.

Furthermore, traffic incidents such as accidents and work zones cause non-recurrent congestion and impact flow dynamics. Drones are able to detect these reductions in capacity by analyzing images of a road segment. The incident information can then be used in the estimation procedure to modify parameters of the traffic model and improve its predictions. It was shown that incorporating the effect of incidents on model parameters improves the traffic state estimates. Moreover, the resulting state variables can be used to recursively update estimates on incident severity based on the deviation of model predictions from observed data [3, 4].

In this thesis, we investigate algorithms for simultaneously estimating incident severity and traffic states using targeted drone observations and measurements from ground sensors. We implement an online algorithm that routes the drone towards observations that minimize the variance on state and parameter estimates. We show that estimating the incident severity in congested conditions using ground sensors only will lead to poor estimates. This results from the inability to distinguish between observation patterns due to congestion and similar patterns due to a reduction in capacity. However, we are capable of routing a drone towards incident prone locations in congested conditions to obtain an accurate representation of the incident severity. The proposed framework is shown in Figure 1.1.

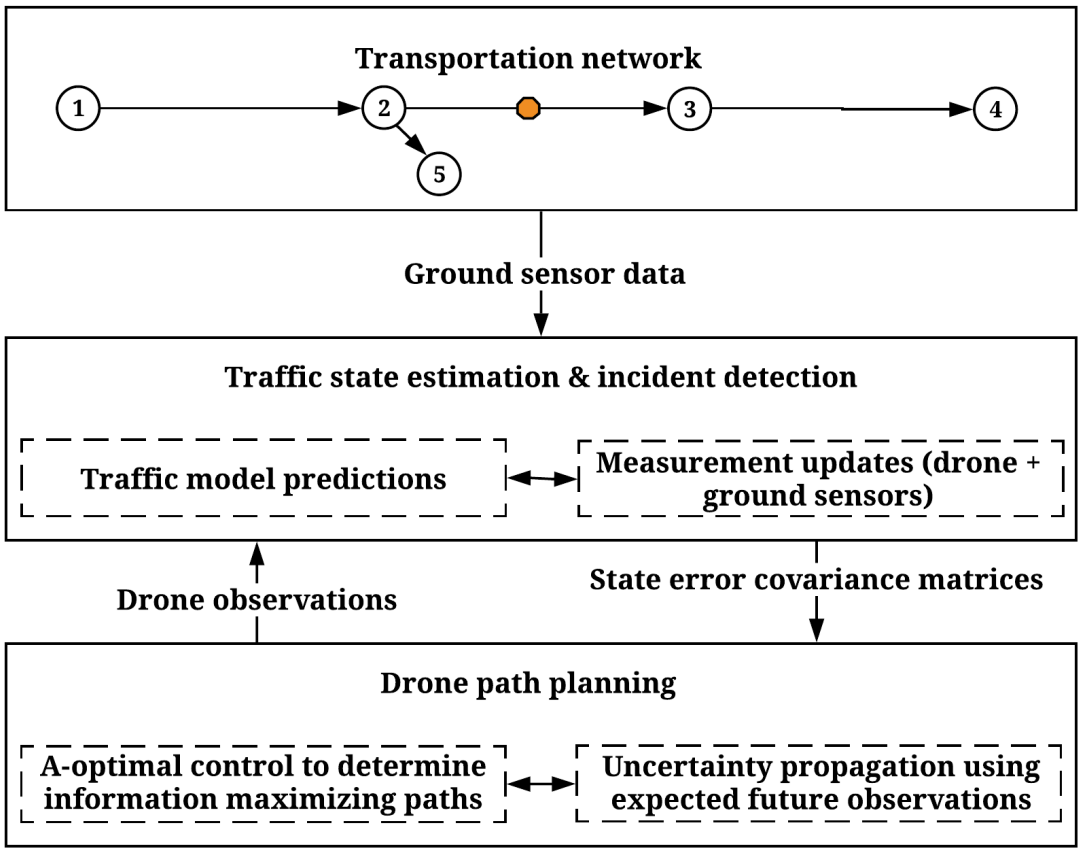

Figure 1.1: Estimation and routing framework to navigate a drone towards informative traffic state and incident observations.

We note that the proposed drone routing framework can be used to improve estimates on traffic states without updating parameters that represent incident severity. In addition, the estimation framework for traffic states and parameters can be implemented using ground sensors solely (i.e., without routing a drone to obtain targeted observation). However, we believe that incorporating incident information in the estimation process and deploying a drone to obtain targeted observations will significantly improve the resulting traffic state and in-

cident severity estimates. While previous studies evaluated the benefit of using incident information to adjust model parameters and obtain improved traffic state estimates [3, 4, 5, 6, 7, 8, 9], we show that generating targeted observations using a drone further improves both state and parameter estimates.

To plan for multiple drones working in parallel, we seek partitioning algorithms that divide the network into subnetworks monitored by specific drones. We aim to minimize the inter-flow between subnetworks to reduce the dependence of estimates in one subnetwork on boundary flows estimated from neighboring subnetworks. In addition, we aim to maintain balanced subnetworks to divide the computational cost across drones. We propose a flow-weighted spectral partitioning algorithm and compare its performance against an agglomerative heuristic. We show that the flow-weighted spectral partitioning method is superior based on the aforementioned metrics.

1.2 Problem Statements & Contributions

Following is a list of questions we aimed to answer in this thesis and the corresponding solution framework:

1. **How will model parameters change as a result of incidents?**

   To recursively estimate the impact of incidents on model parameters, we need to determine how the parameters vary under different levels of incident severity. As free flow speed decreases, the critical density will increase. The capacity can be computed as the product of free flow speed and critical

density. We analyze the relative change in these parameters by fitting speed-density relationships through incident data generated from a microsimulation model.

2. **In case there are incident prone road segments, how can we efficiently determine traffic states and model parameters reflecting incident severity using model-based traffic estimation methods?**

   Estimating model parameters and traffic states can be implemented using various forms of nonlinear filtering techniques (e.g., extended Kalman filter, unscented Kalman filter, ensemble Kalman filter, or particle filters) [10]. We seek a filtering method that is computationally efficient. In addition, we must be able to easily determine the uncertainty on parameters and states from the filtering process. This uncertainty will be required to navigate the drone towards informative observations. We implement a dual state ensemble Kalman filter where the uncertainty is represented by Gaussian distributions and propagated forward in time efficiently using ensemble members.

3. **How can we analyze and quantify the uncertainty on traffic states and model parameters?**

   After determining the uncertainty on states and parameters from the filtering process, we need to translate this uncertainty to a measure that can be used as the drone routing objective. In the proposed estimation and routing framework, we choose the expected variance of the error on the best state or parameter estimate to navigate the drone at every time step. This variance

can be easily computed as the trace of future state error covariance matrices. The future covariance matrices are generated by updating current error statistics using anticipated drone observations.

4. **How can we adaptively navigate a drone in an online setting to minimize the uncertainty on traffic states and model parameters?**
   Given the state error covariance matrices at a particular time step, the next drone move is computed by propagating the state error covariance matrices into the future and computing the expected variance on state errors. This propagation relies on anticipated observations that we expect the drone to obtain in the subsequent time steps. After the drone moves in the variance minimizing direction, we collect actual measurements from drone and ground sensors. These actual measurements are used to update the state error covariance matrices. Thus, the next drone move must be computed at every stage by propagating updated state error covariance matrices. To navigate the drone in this online setting, we propose a one step lookahead algorithm that continuously determines the variance minimizing direction starting from the current drone position and state error covariance matrices.

5. **How can we effectively distribute drones for monitoring a large network?**
   While the proposed estimation-routing framework can be used for a single drone that is monitoring roads with a high functional class, the computational cost and estimation quality will deteriorate on larger networks. One possible approach for estimation over larger networks is to deploy multiple

drones working in parallel. To plan for multiple drones, we need to distribute the drones over subnetworks. We investigate network partitioning methods that aim to minimize inter-flow and maintain balanced subnetworks. Minimizing inter-flow reduces the dependence of estimates in one subnetwork on the quality of estimates in an adjacent subnetwork. Maintaining balanced subnetworks ensures that the computational cost is divided across drones. We implement a flow-weighted spectral partitioning algorithm that is based on the aforementioned objectives. Furthermore, we balance the subnetworks by flow to ensure that the drones are abundantly deployed in congested portions of the network where accurate estimates are needed to inform congestion management strategies.

In summary, the contributions of this thesis include (1) analysis of the variation in model parameters under incidents, (2) a model-based traffic estimation procedure to simultaneously estimate traffic states and model parameters reflecting incident severity, (3) a coupled planning and estimation framework for navigating a drone to improve estimates on traffic states and parameters, and (4) an algorithm for partitioning a network into subnetworks monitored by separate drones.

1.3 Organization

The remainder of this thesis will proceed as follows: Chapter 2 discusses literature relevant to traffic state estimation, incident detection, simultaneous traffic state estimation and incident detection, as well as network partitioning algo-

rithms. Chapter 3 analyzes the impact of incidents on traffic model parameters and presents the coupled estimation-routing framework. Chapter 4 presents the proposed flow-weighted spectral partitioning algorithm for dividing a network into subnetworks monitored by separate drones. Conclusions and future work are discussed in Chapter 5.

Chapter 2

Literature Review

This chapter discusses previous work related to the proposed coupled estimation and routing framework. In Section 1, traditional data-driven methods for incident detection are discussed. Section 2 presents model-based traffic estimation. Section 3 discusses methods for simultaneous incident detection and traffic estimation. In Section 4, literature relevant to network partitioning is discussed.

2.1 Non-Recurrent Congestion & Incident Detection

Non-recurrent congestion is caused by capacity-reducing incidents such as accidents, adverse weather conditions, and work zones. This type of congestion is considered to be the primary source of travel time variability and accounts for up to 30% of congestion delay during peak periods [11, 12]. Traditionally, non-recurrent congestion detection methods rely on comparing expected traffic conditions with sensor measurements using a data-driven approach. These algorithms detect that an incident occurred once collected data significantly deviates from expected conditions [13]. Comprehensive reviews on such incident detection algorithms can be found in [14, 15].

A popular class of data-driven incident detection methods that is referred to as comparative algorithms relies on analyzing occupancy data from an upstream and downstream sensor [14, 16]. This analysis compares observed patterns to spec-

ified thresholds in a decision tree with multiple tests. The tests are used to distinguish between bottleneck congestion and incident patterns. They are also used to determine whether observed patterns are due to incidents or random fluctuations in the data.

Another class of incident detection methods relies on time series analysis to provide traffic forecasts. An incident is detected if the algorithm identifies significant deviations between observed measurements and forecasts [14, 15]. These methods include the standard normal deviation algorithm, the double exponential algorithm, and the autoregressive integrated moving average algorithm (ARIMA). The primary differences between these algorithms are the forecast approach and the test used to determine when observed deviations are significant. To improve the performance of the algorithms under random data fluctuations, low pass filtering was used to smooth raw data within time windows [13, 14].

In the late 1990's, artificial neural networks (ANN) where used for incident detection [17, 18, 19]. In these methods, an ANN is trained with historical or simulated data from detectors. This creates a mapping that determines whether an incident occurred based on observed data.

Alternative data-driven methods include the wavelet-based approach and techniques based on analyzing probe vehicle data [15, 20]. The primary drawbacks of data-driven methods pertain to (1) fitting or specifying a large number of parameters, (2) inability to distinguish between traffic patterns from incidents and similar data due to congestion under normal operating conditions, (3) sus-

ceptibility to random fluctuations in traffic data, and (4) inability to adequately predict traffic states due to the lack of a model representing traffic flow dynamics [6, 14, 15].

2.2 Traffic Estimation

Model-driven traffic state estimation aims to infer state variables (e.g., density, speed, and flow) from partially observed noisy data and a model representing traffic flow dynamics. The traffic model is used to determine traffic states on unobserved road segments and to predict future traffic states. Sensor data is used to adjust model predictions. In certain cases, the observed data is also used to calibrate model parameters. Recent comprehensive reviews on traffic state estimation can be found in [2, 10].

In this thesis, we focus on sequential data assimilation estimation methods. These methods use a nonlinear filter to iteratively update state estimates as data is being collected. The updates typically minimize the mean square error (MMSE) based on the uncertainty that is associated with model predictions and the observation noise.

Generally, the nonlinear filters rely on a state-space model shown in equations 2.1 and 2.2. $\mathbf{x^t}$ is a state vector containing state variables that we desire to estimate. The function f represents a model for the evolution of state variables across time. The system noise is denoted by w, and it reflects the uncertainty on predictions obtained from the aforementioned model. $\mathbf{y^{t+1}}$ are sensor measure-

ments. h is an observation function that translates state variables to corresponding observations. ϵ is the observation noise.

$$\mathbf{x^{t+1}} = f(\mathbf{x^t}) + w \tag{2.1}$$

$$\mathbf{y^{t+1}} = h(\mathbf{x^{t+1}}) + \epsilon \tag{2.2}$$

In traffic state estimation problems, the state variables reflect traffic flow characteristics such as density, speed, and flow. The function f is a traffic model that represents how the state variables change with time based on traffic dynamics. Thus, the objective is to find a best estimate on the traffic states at every time step using a traffic model and sensor observations.

In case of linear functions (f, h) and Gaussian white noise (w, ϵ), the MMSE state estimates can be obtained using equations 2.3–2.7. These equations correspond to the popular Kalman filter [21]. $\mathbf{F}$ and $\mathbf{H}$ are matrices corresponding to the linear transformations f and h, respectively. $\mathbf{W}$ and $\mathbf{E}$ are covariance matrices such that $w \sim \mathcal{N}(0, \mathbf{W})$ and $\epsilon \sim \mathcal{N}(0, \mathbf{E})$. $\mathbf{P^t}$ is the state error covariance matrix. $\mathbf{K^{t+1}}$ is referred to as the Kalman gain.

$$\mathbf{x^{t+1|t}} = \mathbf{F}\mathbf{x^t} \tag{2.3}$$

$$\mathbf{P^{t+1|t}} = \mathbf{F}\mathbf{P^t}\mathbf{F}^T + \mathbf{W} \tag{2.4}$$

$$\mathbf{K^{t+1}} = \mathbf{P^{t+1|t}H}^T(\mathbf{HP^{t+1|t}H}^T + \mathbf{E})^{-1} \tag{2.5}$$

$$\mathbf{x^{t+1}} = \mathbf{x^{t+1|t}} + \mathbf{K^{t+1}}(\mathbf{y^{t+1}} - \mathbf{Hx^{t+1|t}}) \tag{2.6}$$

$$\mathbf{P^{t+1}} = \mathbf{P^{t+1|t}} - \mathbf{K^{t+1}HP^{t+1|t}} \tag{2.7}$$

However, in traffic state estimation, the traffic models used are usually nonlinear functions of state variables. This nonlinearity is due to the difference in flow dynamics between congested and uncongested regimes. In that case, suboptimal nonlinear extensions of the Kalman filter can be used to iteratively update the states and their corresponding statistics [10]. In this thesis, we use the ensemble Kalman filter (EnKF) which will be further discussed in Chapter 3. The EnKF is a stochastic filter that propagates ensemble members. These ensemble members determine the best state estimate and the state error covariance matrix at every time step [22]. Other notable nonlinear filters include the extended Kalman filter (EKF) [23] and the unscented Kalman filter [24, 25]. Compared to the EKF, the EnKF does not require model Jacobian computations.

Alternatively, we can use a sequential Monte Carlo importance sampling approach that approximates the posterior distribution by weighted samples propagated via the state model [26]. This approach is often referred to as the particle filter. While particle filters can properly represent arbitrary (e.g. non-Gaussian) posterior distributions given a sufficient number of particles, they are computationally expensive. In addition, particle filters suffer from sample degeneracy

problems where the normalized weights of almost all particles tend to zero after a few iterations [27].

2.3 Simultaneous Traffic Estimation & Incident Detection

The incident detection algorithms discussed in Section 2.1 are based entirely on analyzing observed data. An alternative class of methods relies on macroscopic traffic modeling to detect incidents and predict the corresponding traffic dynamics. In particular, researchers explored estimation methods that simultaneously estimate traffic states and certain parameters that reflect incident severity. Wang and Papageorgiou [28] proposed an extended Kalman filter that uses a macroscopic traffic flow model to estimate densities as well as calibrate the free flow speed and critical density. They implemented joint state estimation in which parameters and boundary variables are added to the state space, and they considered that flow and mean speed measurements could be obtained. Recent articles on simultaneous estimation of states and fundamental diagram parameters include the use of count and trajectory data in a single optimization framework [29], and a moving horizon approach that determines the traffic state and incident parameters which minimize a quadratic cost function [7].

Alternative estimation techniques include methods that aim to identify the most likely traffic model given observed data. In the case of incident detection, each model represents a different parametrization that reflects a certain level of incident severity. The first article to consider this approach used an extended Kalman filter to select the most likely model [5]. Then, this framework was enhanced to al-

low for dependencies between models chosen across time steps [3]. Specifically, an interactive multiple model ensemble Kalman filter and a multiple model particle filter were developed to simultaneously estimate traffic state and incident severity given specified incident evolution dynamics [3, 4, 6].

For typical sensor data such as speed and occupancy measurements, the aforementioned methods are not able to distinguish between congestion observations due to incidents and similar observations from congested states under normal operating conditions. To address this problem, we propose the use of unmanned aerial vehicles to directly estimate the incident severity and traffic state. Drones can be equipped with cameras to monitor road and traffic conditions [30]. Then, using image processing techniques they can detect incidents by analyzing objects in a frame across time [31, 32].

Integrating drones into traffic estimation methods raises challenges pertaining to the appropriate parameter-state estimation and drone navigation procedures. We assume that density data from loop detectors and less frequent speed measurements from GPS equipped probe vehicles are available. For estimation, we use a dual state-parameter ensemble Kalman filter (EnKF) with traffic dynamics represented by the cell transmission model (CTM). This generates time-varying Gaussian distributions that represent the uncertainty on parameter and state estimates. To minimize this uncertainty, we develop an online one-step lookahead path planning algorithm that evaluates candidate drone trajectories based on anticipated reduction in average variance of state and parameter estimates.

2.4 Network Partitioning

The literature on network partitioning algorithms is extensive. These algorithms can be broadly classified into agglomerative/divisive heuristics, integer programming based approaches, and spectral partitioning algorithms. Integer programming formulations for the partitioning problem are proven to be NP-hard [33] and approximation heuristics have been proposed [33, 34].

Heuristics for generating partitions based on agglomerative and divisive clustering have been recently used in various transportation related applications. Saeedmanesh and Geroliminis [35] used an agglomerative clustering heuristic for generating partitions based on "snake" similarities for applications of the macroscopic fundamental diagram. Etemadnia et al. [33] developed similar heuristics for distributed traffic management. Johnson et al. [36] developed another heuristic for decentralized traffic management. This latter heuristic aims to minimize boundary nodes in subnetworks and to create subnetworks of similar size. The heuristic performed better than the METIS algorithm proposed in [37].

Spectral partitioning is an alternative approach for partitioning a graph [38, 39, 40, 41, 42]. Bell [43] applied a capacity-weighted form of the spectral partitioning methods to investigate network vulnerability. Other transportation applications include air traffic control and urban traffic signal control systems [44, 45]. The partitioning mechanism is based on the eigenvalues associated with the graph Laplacian. The partitions that result from spectral partitioning have low intercluster similarity [39].

Additionally, using the normalized Laplacian in spectral partitioning methods generates graphs that are balanced by weight. This is an important feature since ignoring the balance requirement results in cuts that isolate a small number of peripheral nodes. For example, the minimum cut program that aims to minimize the weight between resulting partitions will often result in separating one node from the rest of the network [41]. Generally, incorporating balance requirements causes cut problems to become NP-hard. Spectral partitioning provides an approximate solution that generates balanced subnetworks [38, 41, 42].

In this thesis, we implement a flow-weighted spectral partitioning algorithms that aims to minimize the inter-flow between subnetworks and to maintain subnetworks balanced by total flow. This network partitioning algorithm can be used to divide a network into subnetworks monitored by separate drones that are working in parallel. We disregard the possibility of information sharing between drones and its implications on drone deployment.

Chapter 3

Traffic State Estimation, Incident Detection, & Drone Navigation

In this chapter, we propose a coupled estimation-routing framework to navigate a drone towards informative observations that minimize the variance on state and parameter estimates. First, we investigate the variation in traffic model parameters under different levels of incident severity. Then, we use a dual state ensemble Kalman filter to simultaneously estimate model parameters and traffic states. This estimation procedure will be integrated into an online planning algorithm that navigates the drone based on the anticipated reduction in uncertainty along candidate drone paths. The estimation-routing framework is shown in Figure 1.1.

3.1 Non-Recurrent Congestion Impact on Traffic Model Parameters

This section presents the cell transmission model. In this thesis, we use the cell transmission model to represent traffic dynamics in the estimation-routing framework. In addition, we analyze the variation in model parameters under incidents. The analysis in this section will be used in Sections 3.2 and 3.3 to determine how model parameters should be updated based on observed sensor data.

3.1.1 Cell Transmission Model

The Lighthill-Whitham-Richards partial differential equation (LWR PDE) is used to represent traffic dynamics. This is shown in Equation 3.1 where $\rho(x,t)$ and $v(\rho(x,t))$ are the density and velocity at a particular point in space and time, respectively. We use a speed-density relationship defined in Equation 3.2 that corresponds to a triangular flow-density diagram. In this equation, ρ_{cr} is the critical density, ρ_j is the jam density, and v_{max} is the free flow speed.

$$\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial(\rho(x,t)v(\rho(x,t)))}{\partial x} = 0 \tag{3.1}$$

$$v(\rho(x,t)) = \begin{cases} v_{max} & \text{for } \rho(x,t) \leq \rho_{cr} \\ \frac{v_{max}\rho_{cr}(\rho_j - \rho(x,t))}{\rho(x,t)(\rho_j - \rho_{cr})} & \text{otherwise} \end{cases} \tag{3.2}$$

The LWR PDE is discretized for practical implementation using a Godunov scheme to obtain the cell transmission model (CTM) [46, 47, 48]. Specifically, time is discretized into time steps of duration Δt, and space is discretized into cells of length Δx such that $\Delta x = v_{max}\Delta t$. Letting $n(x,t)$ be the number of vehicles in cell x at time t and $y(x,t)$ be the number of vehicles entering cell x at the t-th time step, we can approximate $n(x,t)$ and $y(x,t)$ using Equations 3.3 and 3.4. Then, we can determine $y(x,t)$ using Equation 3.5 and $n(x,t)$ using Equation 3.6. The calculated $n(x,t)$ values can be used to track densities in cells across time via Equation 3.3.

$$n(x,t) \approx \rho(x,t)\Delta x \tag{3.3}$$

$$y(x,t) \approx \rho(x,t)v(\rho(x,t))\Delta t \tag{3.4}$$

$$y(x,t) = \min\left\{n(x-\Delta x,t), \frac{\rho_{cr}}{\rho_j - \rho_{cr}}\left(\rho_j \Delta x - n(x,t)\right)\right\} \tag{3.5}$$

$$n(x,t+\Delta t) = n(x,t) + y(x,t) - y(x+\Delta x,t) \tag{3.6}$$

3.1.2 Impact of Incidents on CTM parameters

To investigate the impact of non-recurrent congestion on the speed-density relationship and vehicle throughput, we use the PTV VISSIM microsimulation software with reduced speed zones representing non-recurrent congestion regions [49]. First, we calibrate the speed-density relationship without any incidents for a three lane freeway by fitting Equation 3.2 through data points generated at a particular location along the freeway. The dashed line in Figure 3.1 corresponds to the fitted relationship under normal operating conditions.

Then, we introduce different levels of incident severity using reduced speed zones and determine a least squares fit for a piecewise linear function through the observed speed and density data points. The intersection point of the two line segments indicates the critical density corresponding to a particular incident free flow speed. In Figure 3.1, the intersection point is denoted by star markers. As shown, the critical density increases with reduced speed. The increase in critical density with incident severity is expected since at lower free flow speeds vehicles

can travel with shorter headways before backward shock waves are initiated.

Note that the non-recurrent congestion incidents we consider do not significantly impact the jam density. This type of incidents could represent adverse weather conditions or roadside accidents and work zones. The analysis in this section will be used in Section 3.2.4 to determine how incident parameters should be updated based on observed measurements. Compared to existing methods that determine the change in parameters for lane-blocking incidents using highway capacity manual (HCM) recommendations and a fixed incident free flow speed [3, 4, 6], we aim to develop a relationship that determines the change in critical density as a function of reductions in free flow speed using the observed patterns in Figure 3.1.

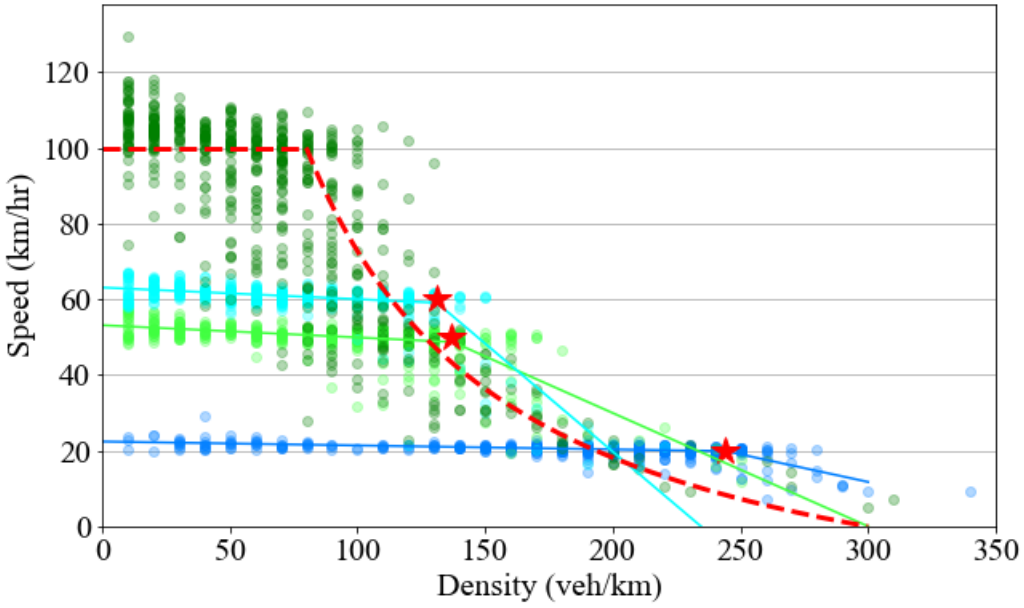

Figure 3.1: Impact of non-recurrent congestion on the speed-density relationship.

3.2 Dual State Ensemble Kalman Filter for Traffic Estimation & Incident Detection

To estimate traffic state and incident severity across time, we use a dual state space ensemble Kalman filter (EnKF) [10, 22, 50]. The traffic state is represented by densities propagated forward using the cell transmission model with additive Gaussian white noise. The incident severity is represented by the free flow speed parameters v_{max} at incident prone locations. These parameters are propagated forward using a random walk. In terms of observations, the traffic state is directly observed using loop detector density measurements while the incident parameter v_{max} is observed using Equation 3.2 given less frequent speed measurements.

3.2.1 Dual State EnKF

The dual state EnKF is composed of two separate ensemble Kalman filters for traffic states and free flow speed working in parallel. Each EnKF is a stochastic filter that propagates ensemble members (samples) representing the state statistics. The filters interact by recursively feeding best estimates into each other at every update step. In particular, the updated parameters are used to adjust the forward model of the traffic state EnKF, and the resulting traffic state estimates inform subsequent parameter updates. In each filter, the ensemble mean is the best estimate on the true traffic state/parameter and the ensemble covariance corresponds to the error on the ensemble mean [22, 50].

While alternative optimization methods could be used to freely specify the

estimation objectives [51], the dual EnKF is a variance minimizing scheme that enables efficient updating of Gaussian covariance matrices which we exploit for drone path planning in Section 3.3. In addition, compared to methods that estimate the most likely traffic model, the dual EnKF uses continuous incident parameters and is thus not limited to a specified set of incident severity levels. Dual estimation procedures are not capable of capturing correlation between the parameters and traffic state variables since they maintain separate filters. However, estimation of parameters based on filtered state estimates may result in smoother parameter estimates [52, 53, 54].

Importantly, the dual estimation procedure allows for integration of data at different time scales to update v_{max} parameters only when speed measurements are collected. Furthermore, using a dual state estimation procedure allows us to maintain separate covariance matrices for states and parameters. Maintaining separate covariance matrices is a critical component of the proposed drone navigation algorithm (Section 3.3) that aims to identify targeted uncertainty minimizing drone measurements.

3.2.2 Traffic State EnKF

The traffic state EnKF is shown in Equations 3.7–3.14 where $\mathbf{k}_j^t$ is a vector of densities corresponding to ensemble member j at time t. $\mathsf{CTM}_{\Delta \mathsf{t}}$ represents forward propagation of densities from time t until time $t + \Delta t$ using Equations 3.3–3.6 and w is Gaussian noise that reflects model errors. For N ensemble members, $\mathbf{A}$ is a matrix in which the columns are the ensemble members, $\mathbf{1_N}$ is an $N \times N$ scale

matrix such that every element is $1/N$, and $\bar{\mathbf{A}}$ is a matrix where every column is the ensemble mean.

In terms of observations, $\mathbf{d_j}$ represents a particular perturbation of the density measurements $\mathbf{d}$ using Gaussian observation errors denoted as ϵ, and $\mathbf{D}$ is a matrix storing the perturbed observations. Υ is a matrix storing the observation perturbations. In addition, $\mathbf{H}$ is an observation matrix which in this case is the identity matrix since state density variables are observed directly.

The updated ensemble members $\mathbf{k}_j^{t+\Delta t}$ are stored in columns of matrix $\mathbf{A^a}$ and $\mathbf{P}$ is the updated ensemble covariance matrix. This analysis procedure is a variance minimizing scheme that enables nonlinear propagation of error statistics and iteratively computes the best state estimates. Compared to the extended Kalman filter, this procedure does not require linearizations and efficiently propagates the state error covariance matrix using ensemble members [22].

$$\mathbf{k}_j^{t+\Delta t|t} = \mathsf{CTM}_{\mathbf{\Delta t}}[\mathbf{k}_j^t] + w \quad \forall j \in \{1, .., N\} \tag{3.7}$$

$$\mathbf{A} = (\mathbf{k}_1^{t+\Delta t|t}, \mathbf{k}_2^{t+\Delta t|t}, ..., \mathbf{k}_N^{t+\Delta t|t}) \tag{3.8}$$

$$\bar{\mathbf{A}} = \mathbf{A1_N} \tag{3.9}$$

$$\mathbf{d_j} = \mathbf{d} + \epsilon_j \quad \forall j \in \{1, .., N\} \tag{3.10}$$

$$\mathbf{D} = (\mathbf{d_1}, \mathbf{d_2}, ..., \mathbf{d_N}) \tag{3.11}$$

$$\Upsilon = (\epsilon_1, \epsilon_2, ..., \epsilon_N) \tag{3.12}$$

$$\mathbf{A^a} = \mathbf{A} + (\mathbf{A} - \bar{\mathbf{A}})(\mathbf{A} - \bar{\mathbf{A}})^T\mathbf{H}^T(\mathbf{H}(\mathbf{A} - \bar{\mathbf{A}})(\mathbf{A} - \bar{\mathbf{A}})^T\mathbf{H}^T + \Upsilon\Upsilon^T)^{-1}(\mathbf{D} - \mathbf{HA}) \tag{3.13}$$

$$\mathbf{P} = \frac{1}{N-1}(\mathbf{A^a} - \mathbf{A^a 1_N})(\mathbf{A^a} - \mathbf{A^a 1_N})^T \tag{3.14}$$

3.2.3 Free Flow Speed EnKF

The ensemble Kalman filter for estimating the free flow speeds at incident prone locations v_{max} follows a similar procedure to the densities EnKF. The parameters are propagated through a random walk as shown in Equation 3.15 where $\mathbf{v}_j^t$ is an ensemble member containing the free flow speeds at incident prone locations. The random walk reflects the change in incident severity across time.

$$\mathbf{v}_j^{t+\Delta t|t} = \mathbf{v}_j^t + w \tag{3.15}$$

The primary difference between the traffic state EnKF and free flow speed EnKF is the nonlinear observation operator. Specifically, if we consider that the critical density ρ_{cr} is a function of the free flow speed v_{max} as shown in Figure 3.1, then it would not be possible to create the linear observation matrix $\mathbf{H}$. Ignoring the dependence of ρ_{cr} on v_{max} and given assimilated densities from the traffic state

EnKF, we can construct time varying observation matrices that correspond to the observed velocities using Equation 3.2. However, this leads to a poor performance of the filter.

To incorporate nonlinear observations, we introduce diagnostic variables that represent model predicted measurements into the state space. In particular, we introduce the matrix $\hat{\mathbf{A}}$ such that its columns are predicted velocities computed through a nonlinear function, $\mathsf{M}(\cdot)$, which represents Equation 3.2 and accounts for the dependency of ρ_{cr} on v_{max}. Subsequently, the ensemble update equation is replaced with Equation 3.17 [50].

$$\hat{\mathbf{A}} = (\mathsf{M}(\mathbf{v}_1^{t+\Delta t|t}), \mathsf{M}(\mathbf{v}_2^{t+\Delta t|t}), ..., \mathsf{M}(\mathbf{v}_N^{t+\Delta t|t})) \tag{3.16}$$

$$\mathbf{A^a} = \mathbf{A} + (\mathbf{A} - \bar{\mathbf{A}})(\hat{\mathbf{A}} - \hat{\mathbf{A}}\mathbf{1_N})^T((\hat{\mathbf{A}} - \hat{\mathbf{A}}\mathbf{1_N})(\hat{\mathbf{A}} - \hat{\mathbf{A}}\mathbf{1_N})^T + \Upsilon\Upsilon^T)^{-1}(\mathbf{D} - \hat{\mathbf{A}}) \tag{3.17}$$

3.2.4 Model Parameter Updates under Non-Recurrent Congestion

The procedure for introducing diagnostic variables to address nonlinearities works well in situations where the function $\mathsf{M}(\cdot)$ is monotonic and not highly nonlinear [50]. If $\mathsf{M}(\cdot)$ is non-monotonic, then it would not be clear if the updates should increase or decrease v_{max} with a change in observed speed. This has implications on specifying the dependence of ρ_{cr} on v_{max}.

To maintain monotonic $\mathsf{M}(\cdot)$ functions, we consider that as v_{max} varies, the backward wave speed remains fixed at the calibrated value in incident free conditions. Graphically, this corresponds to updating the fundamental diagrams as shown using dashed lines in Figure 3.2. The updated critical density ρ_{cr}^t for any value v_{max}^t is then given by Equation 3.18 where v_{max}^0 and ρ_{cr}^0 are the original calibrated parameters under incident-free conditions. From Figure 3.1, we assume that the star markers will lie on the dashed line, which we believe is a reasonable assumption. We note that if the parameters are otherwise updated such that at any stage the backward wave speed increases when v_{max} decreases or vice versa, the resulting $\mathsf{M}(\cdot)$ functions will be non-monotonic.

$$\rho_{cr}^t = \frac{\rho_{cr}^0 v_{max}^0 \rho_j}{v_{max}^t(\rho_j - \rho_{cr}^0) + \rho_{cr}^0 v_{max}^0} \tag{3.18}$$

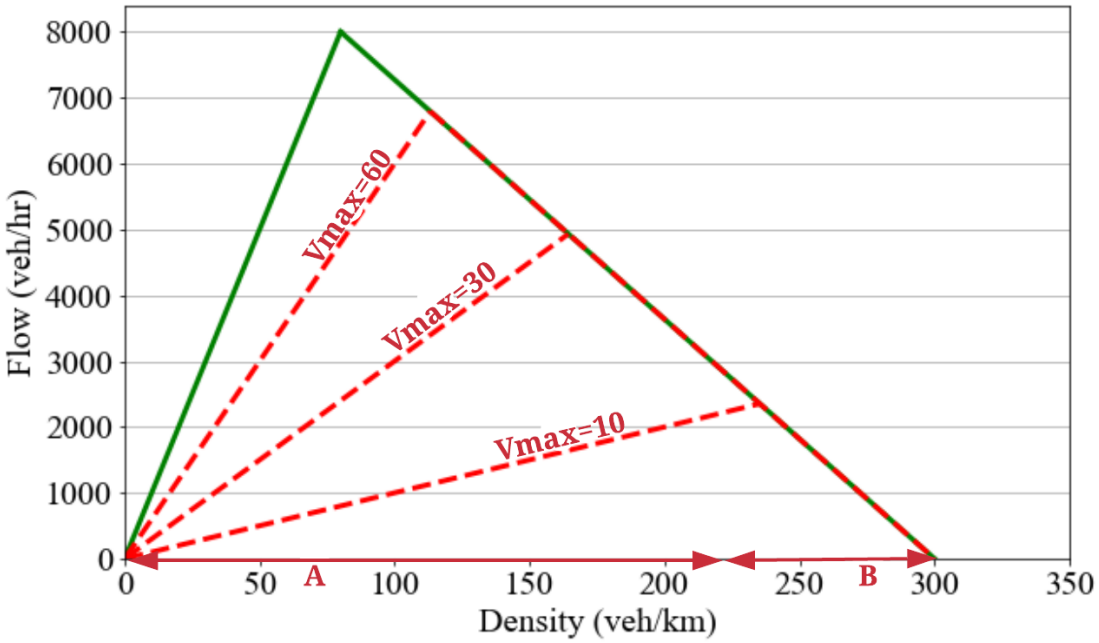

Figure 3.2: Change in fundamental diagram with increasing incident severity. Dashed lines correspond to incident fundamental diagram.

3.2.5 The Dual State EnKF Algorithm

Therefore, in the dual EnKF estimation procedure, we iteratively update traffic state estimates using Equations 3.7–3.14. Once we observe speed measurements, we use the ensemble mean of the assimilated densities for updating the free flow speed parameters via an EnKF approach for handling nonlinear observations and Equation 3.2. Then, the ensemble mean of the v_{max} EnKF is used to update the parameters v_{max} and ρ_{cr} in the $\text{CTM}_{\Delta t}$ forward model until future velocity measurements are obtained. This dual EnKF procedure is shown in Algorithm 1.

Algorithm 1 Dual EnKF for traffic states and free flow speed parameters at incident prone locations

Initialization:

(1) Define $\mathsf{CTM}_{\Delta t}$ based on incident-free calibrated parameters

(2) Create initial ensembles for densities across cells

(3) Create initial ensembles for free flow speeds at incident prone locations

Dual EnKF:

for $time$ in $estimation\ horizon$ **do**

(4) Propagate density ensembles forward using Equations 3.3–3.6

(5) Update ensembles using densities EnKF through Equations 3.7–3.14

if get speed observation **then**

(6) Propagate free flow speeds ensembles using Equation 3.15

(7) Use mean of densities ensemble, Equations 3.2, 3.7–3.12, and 3.14–3.18 to update ensembles via free flow speeds EnKF

(8) Update v_{max} and ρ_{cr} parameters in $\mathsf{CTM}_{\Delta t}$ using mean of free flow speeds EnKF and Equation 3.18

3.3 Coupled Estimation & Drone Navigation

The proposed dual state estimation procedure in Section 3.2 can efficiently estimate states and parameters in situations where the densities vary significantly during the estimation time horizon. In addition, given low values of density as shown in region A of Figure 3.2, we can effectively distinguish between speed observations that we expect the different fundamental diagrams to generate.

However, in region B of Figure 3.2, the fundamental diagrams coincide and can not be distinguished from speed and density observations. This is expected in any estimation procedure that attempts to estimate the free flow speed or capacity from speed and density observations. Given high densities and low speed measurements, we would not be able to determine whether the observations correspond to congested conditions in an incident-free fundamental diagram or if there is a reduction in physical capacity and free flow speed. In terms of the dual EnKF method, this would results in a high variance on the parameter estimates.

To address this estimation problem in region B, we propose the use of unmanned aerial vehicles to directly estimate incident severity and traffic state. Drones can be equipped with cameras to monitor road and traffic conditions [30]. Then, using image processing techniques they can detect incidents by analyzing objects in a frame across time [31, 32].

The use of drones for traffic estimation raises challenges pertaining to the appropriate drone navigation procedure. At any particular drone location, the next drone move is determined by finding the path that minimizes the anticipated fu-

ture variance on the state errors. To minimize this variance in an online setting where the state covariance matrices are continuously updated, we develop a one-step lookahead path planning algorithm.

3.3.1 A-optimal Control Trajectory Planning Objective

The drone can collect accurate density measurements and direct v_{max} observations up to observation errors. To navigate the drone towards informative observations, we implement the coupled estimation and path planning framework shown in Figure 1.1. In this framework, we are not only determining the best traffic and parameter estimates, but also identifying the set of information maximizing anticipated drone observations. The control protocol is an instance of A-optimal control [55, 56, 57].

Consider that $\hat{\psi}_p^{t+\Delta T}$ is the best traffic state or parameter estimate obtained after ΔT drone observations into the future along path p. The expected variance of the error on the state estimate is equivalent to the trace of the propagated error covariance matrix $\mathbf{P}$ as shown in Equation 3.19. We use this as a measure of future uncertainty $J_p^{t+\Delta T}$ that results after the drone traverses path p. Then, we aim to determine the path p^* that minimizes this uncertainty measure among the set of m candidate paths $\{p_1, .., p_m\}$ as shown in Equation 3.20.

$$\begin{aligned} J_p^{t+\Delta T} &= \mathbb{E}[||\psi_p^{t+\Delta T} - \hat{\psi}_p^{t+\Delta T}||_2^2] = \\ &\quad \mathsf{tr}(\mathbb{E}[(\psi_p^{t+\Delta T} - \hat{\psi}_p^{t+\Delta T})(\psi_p^{t+\Delta T} - \hat{\psi}_p^{t+\Delta T})^T]) = \mathsf{tr}(\mathbf{P}) \end{aligned} \tag{3.19}$$

$$p^* = \text{argmin}_{p_1,..,p_m} J_p^{t+\Delta T} \tag{3.20}$$

Note that to compute $\hat{\psi}_p^{t+\Delta T}$ and $\mathbf{P}$ (uncertainty propagation), we need to embed an EnKF that propagates ensemble members at the current time t into the future $t + \Delta T$ using anticipated drone density observations along path p. Therefore, the framework in Figure 1.1 is composed of (1) a global dual state EnKF that updates state error covariance matrices at every time step using actual drone and ground sensor measurements, and (2) multiple dual state EnKFs that are initiated at every time step to propagate current ensemble members into the future based on anticipated drone measurements along each path. This allows us to determine the reduction in uncertainty along each candidate path and the path p^* that minimizes future uncertainty.

The main concern regarding trace minimization methods is that the objective $J_p^{t+\Delta T}$ depends on the magnitude of the state variables [57]. Specifically, the units chosen to represent densities and speed impact the resulting uncertainty measure. However, by maintaining separate state error covariance matrices using the dual state EnKF, we are able to set the objective as a weighted sum of the trace matrices. The weights account for the differences in scale between densities and free flow speeds. Furthermore, the weights can be used to represent the importance of minimizing the uncertainty on parameters relative to traffic states.

We also normalize for the number of traffic state variables in the densities EnKF, K, and the number of parameters in the free flow speeds EnKF, V, to ensure

that the resulting uncertainty measures are comparable. If we denote $J^{t+\Delta T}_{p,v_{max}}$ as the trace of the free flow speeds state error covariance matrix after ΔT observations along path p , $J^{t+\Delta T}_{p,\rho}$ to be the corresponding measure for traffic densities, and λ to be the weighting factor, we can formulate the objective $J^{t+\Delta T}_{p}$ over possible drone paths as shown in Equation 3.21.

$$J^{t+\Delta T}_{p} = \frac{\lambda}{V} J^{t+\Delta T}_{p,v_{max}} + \frac{(1-\lambda)}{K} J^{t+\Delta T}_{p,\rho} \tag{3.21}$$

3.3.2 Online Approximation & Optimization

Once the drone moves along the variance minimizing path p^*, it feeds accurate density measurements and direct v_{max} observations to the global dual state EnKF. Then, the global dual state EnKF updates the state error covariance matrices based on observations from all available sensors and drone measurements. Since the drone position and state error covariance matrices are continuously updated, the resulting uncertainty measure $J^{t+\Delta T}_{p}$ dynamically changes. Specifically, to calculate $J^{t+\Delta T}_{p}$, the state error covariance matrix after ΔT time steps should be obtained by propagating the current error statistics along candidate paths using the embedded EnKFs. Therefore, $J^{t+\Delta T}_{p}$ must be re-calculated using the updated state error covariance matrices and drone position.

To control the drone in this online setting, we use a one step lookahead pol-

icy where at each time step we move the drone in the direction of the path that minimizes an approximation of the cost function $J_p^{t+\Delta T}$ [58]. The approximate cost function, $\tilde{J}_p^{t+\Delta T}$, is based on the assumption that if a drone follows a particular path it would remain moving in the same direction along that path for the entire duration ΔT. In other words, we disregard the possibility of hovering or backtracking.

Finally, to compute $\tilde{J}_p^{t+\Delta T}$, we need to define the anticipated drone observations along candidate paths. We can determine the drone location at every time step in ΔT using the drone speed and the assumption that the drone remains moving in the same direction along a path. We also assume that the drone can observe a specified length of the road underneath its location. Then, for every drone location and time step in ΔT, we consider that the density observations at the drone location will be equal to the mean of density ensembles propagated by the forward model. In other words, the cell transmission model $\mathsf{CTM}_{\Delta \mathsf{t}}$ is used to propagate the density ensembles at time t up to the desired time step, and the mean of the propagated ensembles is considered to be the future density measurements. As for the free flow speed parameters, we consider that the drone can directly observe the true values up to observation errors once it reaches an incident prone segment.

3.3.3 Estimation & Drone Path Planning Algorithm

Therefore, given current density and free flow speed ensembles from the global dual state EnKF, for every path we propagate the density ensembles forward using the cell transmission model to determine expected density observa-

tions as the drone moves along the path. Assuming that the drone will observe the mean of propagated ensembles and the true road condition, we run the embedded dual EnKF for ΔT steps into the future to generate the state error covariance matrices that would be obtained if the drone traverses each path. Then, we calculate $\tilde{J}_p^{t+\Delta T}$ the trace of the covariance matrix after ΔT steps along every candidate path p. We identify the path which minimizes $\tilde{J}_p^{t+\Delta T}$, and move the drone one step in the direction of that path.

Subsequently, once the drone moves in the direction of the path, we receive actual observations from loop detectors, probe vehicles, and drone observations. We assimilate these observations using the global dual state EnKF. This assimilation updates the state error statistics and the parameters in the forward $\mathsf{CTM}_{\Delta t}$ model. The process is then repeated to determine the next drone step using the updated ensembles and new drone position. The procedure is shown in Algorithm 2.

Algorithm 2 On-line algorithm for drone routing and estimation using the dual EnKF

Initialization:

(1) Define $\mathsf{CTM}_{\Delta \mathsf{t}}$ based on incident free calibrated parameters

(2) Create initial ensembles for densities across cells

(3) Create initial ensembles for free flow speeds at incident prone locations

(4) Set initial drone location

Dual EnKF & drone routing:

for $time$ in $estimation\ horizon$ **do**

(4) Propagate density ensembles forward using Equations 3.3–3.6

(5) Update density ensembles using the global dual state EnKF through Equations 3.7–3.14 (include drone observations)

if get speed observation **or** drone at incident prone location **then**

(6) Propagate free flow speeds ensembles using Equation 3.15

(7) Use mean of densities ensemble, Equations 3.2, 3.7–3.12, and 3.14–3.18 to update free flow speeds ensembles using the global dual state EnKF (include drone observations)

(8) Update v_{max} and ρ_{cr} parameters in $\mathsf{CTM}_{\Delta \mathsf{t}}$ using mean of free flow speeds ensembles and Equation 3.18

(9) Generate possible drone paths from current location

(10) For ΔT time steps, determine drone location at every time step along each path

(11) Generate predicted future drone density observations by propagating current density ensembles forward using $\mathsf{CTM}_{\Delta \mathsf{t}}$ along each path

(12) Assuming drone will observe simulated density observations and actual free flow speeds at incident prone locations, use embedded dual state EnKF for ΔT steps into the future to determine $\tilde{J}_p^{t+\Delta T}$ along each candidate path

(13) Move drone one step in direction of p^* (the path with minimum $\tilde{J}_p^{t+\Delta T}$)

3.4 Estimation & Drone Navigation Results

As a proof of concept, we implement in this section the proposed algorithms (Algorithm 1 in Section 3.2 and Algorithm 2 in Section 3.3) on a freeway with an off-ramp modeled in VISSIM [49]. The freeway length, drone starting position, and incident prone locations are shown in Figure 3.3. In this figure, clouds indicate incident locations and the middle circle indicates the drone starting position.

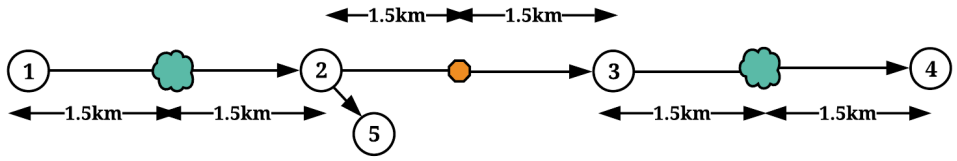

Figure 3.3: VISSIM test network.

In VISSIM, non-recurrent congestion at incident prone location is modeled using a reduced speed zone where the maximum speed is set in the range of 20 km/hr. The inflow demand at node 1 in Figure 3.3 is set at $6,600$ veh/hr. At node 2, half of the demand continues on to node 4 while the other half takes the off-ramp. We use these model parameters to simulate congested conditions upstream (region B in Figure 3.2) and uncongested conditions downstream (region A in Figure 3.2). In terms of observations, we consider that loop detectors feed density measurements at every time step (10 seconds), and that speed measurements are collected from GPS equipped probe vehicles every 30 time steps (5 minutes).

For Algorithm 2, we set $\lambda = 0.5$ to represent equal weights for traffic state and parameter uncertainty measures. We determine ΔT dynamically as the num-

ber of time steps until the drone reaches node 1 if it is traveling upstream, or the number of time steps until it reaches node 4 if it is traveling downstream. We assume that the drone can observe 250 meters at every time step.

3.4.1 Traffic State Estimation & Incident Detection

Algorithms 1 and 2 were both able to accurately estimate densities along the freeway. As shown in Figures 3.4, the upstream density at the incident prone location was estimated to be around 200 veh/km. The downstream density estimates at the incident location varied significantly during the estimation time horizon as shown in Figure 3.5. In these figures, the solid lines indicate true density data from VISSIM, the dotted lines indicate the dual EnKF estimates (Algorithm 1), and the dashed lines indicate UAV-EnKF estimates (Algorithm 2).

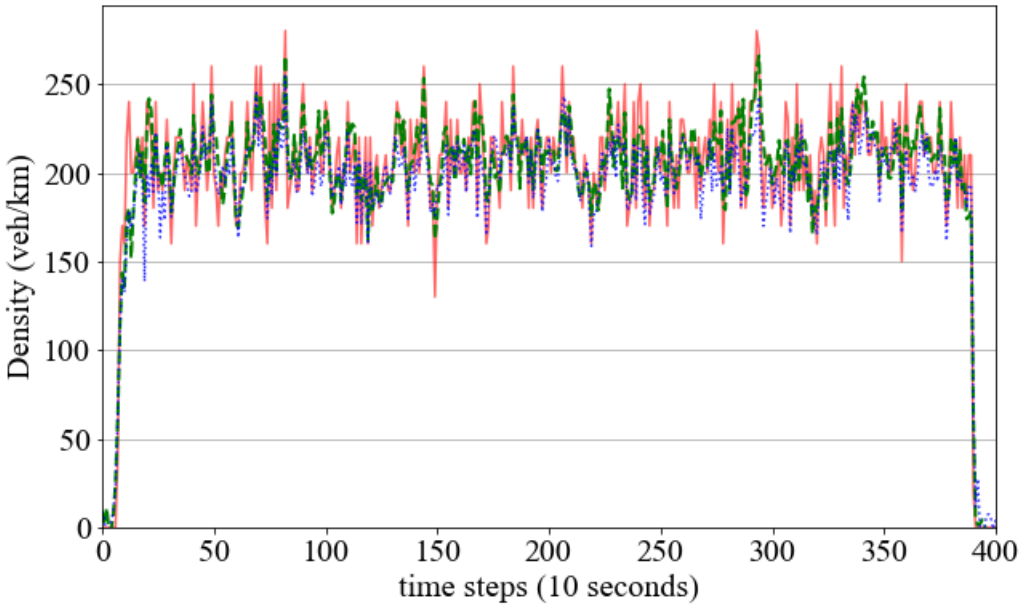

Figure 3.4: Estimated density at the upstream incident location (congested).

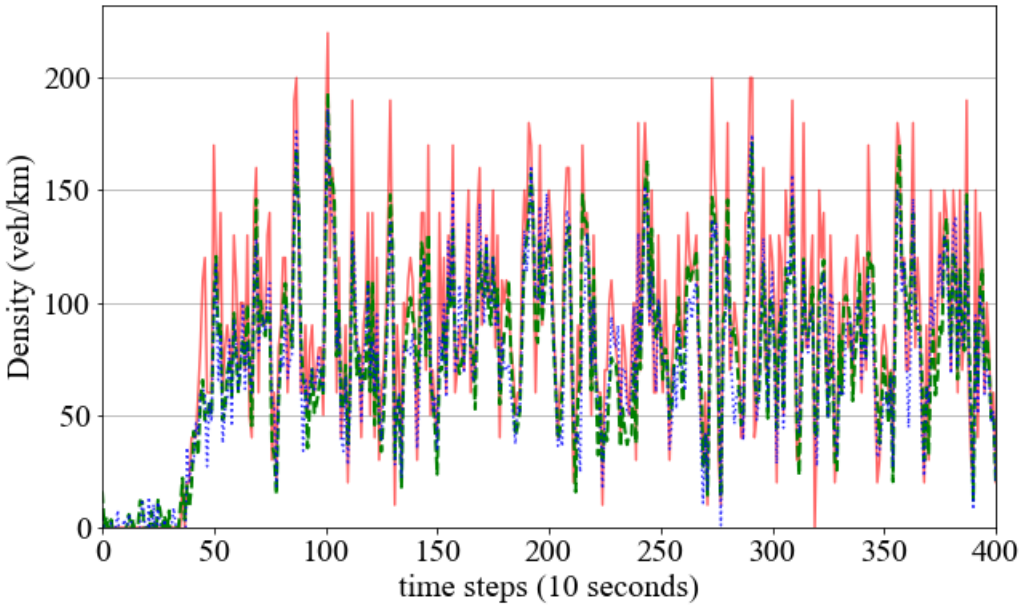

Figure 3.5: Estimated density at the downstream incident location (uncongested).

In terms of free flow speed estimation, the performance of Algorithm 1 differs significantly between uncongested and congested conditions. Specifically, as shown in Figure 3.7, in uncongested conditions the dashed line corresponding to Algorithm 1 quickly converges to the true free flow speed at 20 km/hr. However, as shown in Figure 3.6 corresponding to congested conditions at the upstream incident location, Algorithm 1 is not able to identify the true free flow speed since it is operating in region B of Figure 3.2. In these figures, the dashed horizontal line at 20km/hr indicates ground truth, dashed lines indicate dual EnKF estimates, and solid line indicates UAV-EnKF estimates.

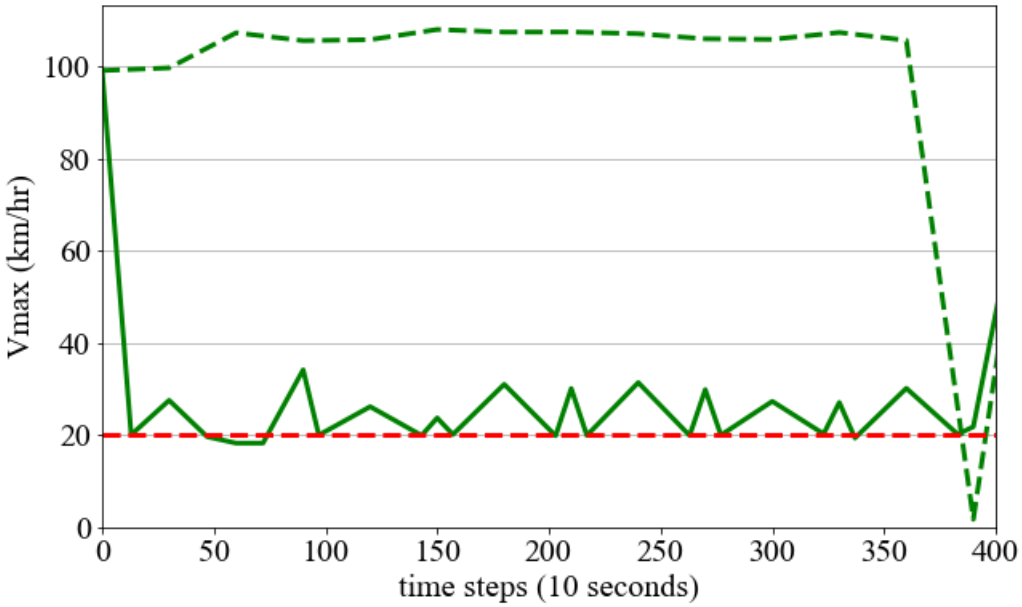

Figure 3.6: Estimated free flow speed at the upstream incident location (congested).

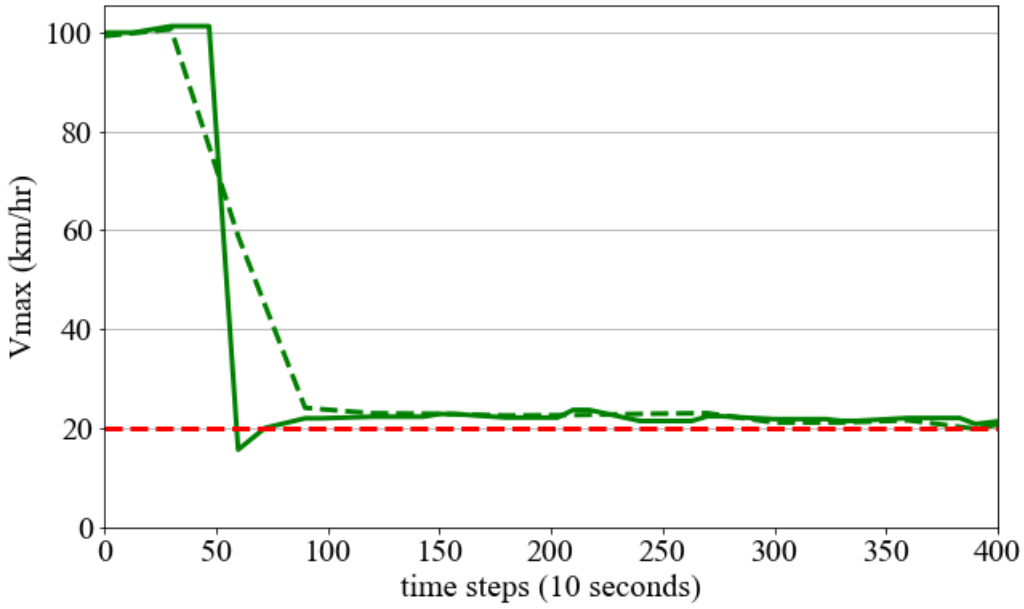

Figure 3.7: Estimated free flow speed at the downstream incident location (uncongested).

When a drone is introduced to aid in the estimation process, we can observe that the estimates of Algorithm 2 in Figure 3.6 (solid line) quickly converges to the true value of the free flow speed. This is a result of the targeted v_{max} observations collected by the drone. In particular, since the upstream incident prone location is congested, the initial v_{max} estimates at that location will have a high variance. This variance prompts the drone to navigate in the upstream direction (path from center to node 1). Once the drone arrives at the upstream location, it directly determines v_{max} up to observation errors using image processing techniques.

We can observe from Figures 3.6 and 3.7 that the UAV-EnKF parameter estimates in uncongested conditions are smoother than the corresponding estimates in congested conditions. This reflects the high variance in congested conditions. Specifically, in congested conditions, a wide range of free flow speeds may generate the speed and density measurements obtained from ground sensors. Therefore, even after the drone determines that the free flow speed is in the range of 20km/hr, subsequent ground sensor measurements are not informative and will result in noisy estimates reflecting the range of possible v_{max} values. However, in uncongested conditions, ground sensors and drone observations agree that v_{max} is in the range of 20km/hr up to observation errors.

3.4.2 Drone Path Planning

To investigate the planning process resulting from Algorithm 2, we plot the drone movement on the freeway across time in Figure 3.8. We can observe that the drone first moves to the upstream incident location since the variance on the upstream free flow speed is higher than the corresponding downstream variance. Once the drone observes the true conditions at the upstream incident location, it reverses direction towards the downstream incident. However, since the v_{max} parameters are propagated through a random walk and the congested conditions at the upstream incident limit the ability to identify the true parameter, the drone decides to go back to the upstream incident location before it reaches the downstream incident. This is expected since as shown in Figure 3.7 the dual EnKF without drone can effectively estimate the true parameter at the uncongested downstream location.

After obtaining 2 observations at the congested upstream location, the drone moves to the downstream incident and continues to traverse the freeway as shown in Figure 3.8. We note that the drone spends 62% of the total estimation time between the middle starting position and the upstream incident prone location. This further shows the need for drone observations in congested conditions that correspond to region B of Figure 3.2.

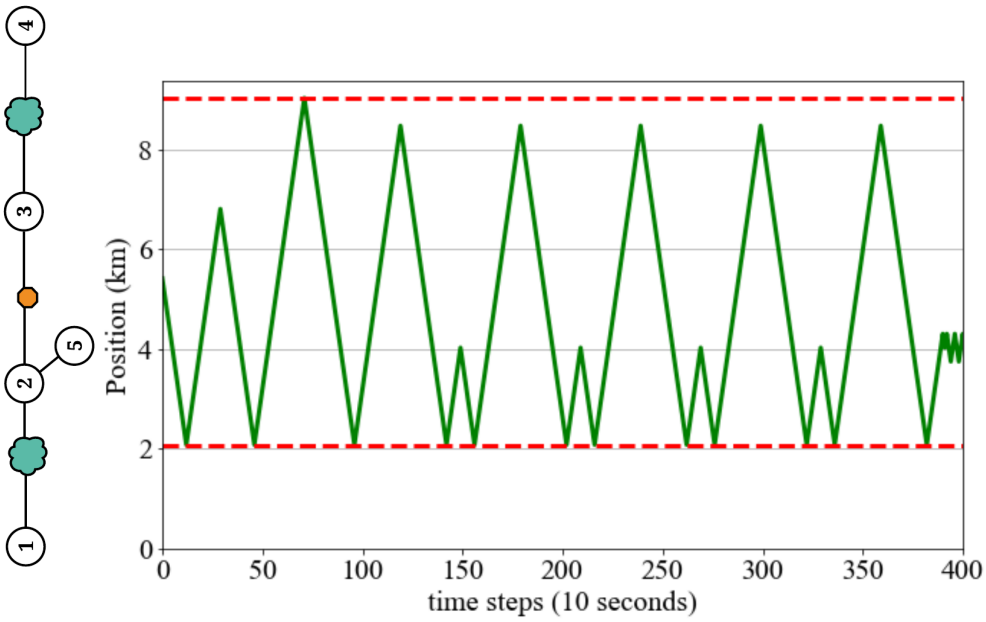

Figure 3.8: Drone movement across time. Dashed lines indicate incident locations.

3.5 Conclusions

In this chapter, we have presented the estimation-routing framework shown in Figure 1.1. First, we analyzed the variation in traffic model parameters under different levels of incident severity. Then, we presented a dual state EnKF for simultaneous estimation of traffic states and incident parameters. We integrated this dual state EnKF into an online planning algorithm to develop the proposed estimation-routing framework.

Subsequently, we showed the advantage of using a drone to obtain targeted variance minimizing observations. In particular, we showed that estimation using ground sensors only in congested conditions will lead to poor estimates on the incident severity. This inability to estimate underlying road conditions is due to the similarity between observation patterns from incidents and corresponding patterns from congestion under normal operating conditions. However, once a drone is used to aid in the estimation process, the drone will navigate towards incident prone locations with high uncertainty on incident parameters. This navigation process will lead to a reduction in the uncertainty since the drone can obtain accurate observations on incident severity using image processing techniques.

Chapter 4

Planning for Multiple Drones Working in Parallel

The results in Chapter 3 indicate that proposed drone estimation-routing framework in Figure 1.1 is capable of accurately estimating incident parameters in situations where this is not possible using ground sensors only. To determine the drone trajectory using Algorithm 2, we analyze at every drone location the anticipated observations along candidate paths. In larger networks, analyzing candidate paths will be computationally expensive. This hinders the applicability of the estimation-routing framework for real-time incident detection.

One approach to reduce the computations required by a drone is to deploy multiple drones working in parallel. However, we need to strategically divide the network across drones to distribute the computational cost and ensure accurate traffic state estimates. In this chapter, we investigate network partitioning methods that aim to minimize inter-flow and maintain balanced subnetworks. In addition, we balance the subnetworks by flow to ensure that the drones are abundantly deployed in congested portions of the network. These objectives (1) reduce the dependence of estimates in one subnetwork on the quality of estimates in an adjacent subnetwork and (2) divide the computational cost across drones. We propose a flow-weighted spectral partitioning algorithm and compare its performance against an agglomerative heuristic. We ignore the possibility of information sharing and its impact on drone deployment and operation. In other words, we assume that the drones are independently monitoring the network.

4.1 Spectral Partitioning

Spectral partitioning is an approximate method for obtaining a cut with minimal cut cost while satisfying balance requirements [38, 41, 42]. This method relies on spectral graph theory to study network properties using the graph Laplacian. Specifically, the eigenvalues and eigenvectors of the Laplacian matrix are analyzed to identify low cost graph cuts. The cost of a cut is defined as a ratio of the weights on cut links to the size of the smaller subnetwork separated by the cut [39, 41].

We consider a directed network G defined by a set of edges A. We define the weighted adjacency matrix M_G^D with elements equal to $w_{(i,j)}$ if link $(i,j) \in A$ and zero otherwise, where $w_{(i,j)}$ is the weight assigned to link (i,j). In this thesis, we will let $w_{(i,j)}$ be the flow on link (i,j) in the flow-weighted spectral partitioning algorithm.

To construct the graph Laplacian, we use an undirected version of M_G^D, denoted by M_G, defined as the sum of M_G^D and its transpose. The graph diagonal matrix D_G is defined as a diagonal matrix where the principal diagonal elements in row i correspond to the sum of elements in row i of M_G. Then, the graph Laplacian, L_G, can be computed using Equation 4.1.

$$L_G = D_G - M_G \tag{4.1}$$

The eigenvalues of an undirected graph Laplacian are real since the matrix is symmetric [38]. Let φ represent the eigenvectors and λ the eigenvalues. The relation between the eigenvectors and eigenvalues for the graph Laplacian is shown in Equation 4.2. According to Spielman [38], the eigenvalues can be defined using Equation 4.3, where S is a vector space of dimensions i, and i is the index of eigenvalue λ_i arranged in an ascending order. The eigenvector for the corresponding eigenvalue can be found using Equation 4.4. The Laplacian matrix L_G is also positive definite and thus the eigenvalues are non-negative.

$$L_G\varphi_i = \lambda_i\varphi_i \tag{4.2}$$

$$\lambda_i = \min_{S\,of\,dim\,i} \max_{x \in S} \frac{x^T L_G x}{x^T x} \tag{4.3}$$

$$\varphi_i = \arg \min_{S\,of\,dim\,i} \max_{x \in S} \frac{x^T L_G x}{x^T x} \tag{4.4}$$

The second smallest eigenvalue and associated eigenvector obtained from Equations 4.3 and 4.4 can be used to partition the graph. The resulting partition is an approximation of the cut that minimizes the ratio cut in Equation 4.5, where $cut(A, \bar{A})$ is the sum of the weights on the links separating the subnetworks A and $\bar{A}$ that are generated from the cut. The denominator of the ratio cut is the size of the smaller subnetwork A, where the size is determined by the number of nodes in A. Minimizing the ratio cut aims to find a cut with minimal weights on the links

separating the subnetworks, and to maintain a balance in size of the generated subnetworks [41].

$$\text{ratio cut} = \frac{cut(A, \bar{A})}{|A|} \tag{4.5}$$

4.2 Flow-Weighted Spectral Partitioning

To minimize inter-flow, we use a flow weighted version of the Laplacian such that the cut cost in Equation 4.5 represents the inter-flow between subnetworks. This will reduce the dependence of estimates in a subnetwork on estimates in adjacent subnetworks. For example, consider a drone operating over a subnetwork where the boundary input flows constitute a significant portion of the flows within the subnetwork. Then, flow estimates within the subnetwork will depend on the accuracy of estimation in adjacent subnetworks. If the boundary flows have a high variance due to uncertainty on incident severity in adjacent subnetworks, the drone will not be capable of accurately estimating the true traffic states within its subnetwork. On the other hand, if the majority of flows in a subnetwork originate within the subnetwork, the drone can properly estimate traffic flows knowing the input demand values.

We also normalize the Laplacian matrix using Equation 4.6 similar to [38, 40, 41, 42]. This normalization will generate partitions that are balanced by the

total flow within the partitions instead of the number of nodes in Equation 4.5. In the estimation-routing framework, balancing the partitions by flow would reduce the per iteration computation time needed to determine the drone trajectory. We prefer to balance the subnetworks by flow instead of number of nodes since the number of nodes is arbitrary and depends on the level of detail in the network model. In the extreme case, consider a highly detailed portion of the network with minimal vehicular traffic. If we partition the network based on number of nodes, we would assign multiple drones to that region while disregarding (less detailed) congested freeways on which accurate estimation and incident detection is needed.

$$L_{symm} = D_G^{-1/2} L_G D_G^{-1/2} \tag{4.6}$$

After calculating the second smallest eigenvalue and associated eigenvector of the normalized Laplacian, the nodes of the network are sorted based on the magnitude of the corresponding element in the eigenvector. The sorted list of nodes is then divided into two parts based on the signs of the corresponding eigenvector elements. This will generate the required partitions [42, 43]. The full algorithm for the flow weighted spectral partitioning is shown in Algorithm 3.

Since the spectral partitioning method proposed is based on link flows, a few implementation issues need to be considered. The use of the second smallest eigenvalue as the basis for partitioning requires the graph to be connected. Specifically, the weighted adjacency matrix M_G should result in a connected graph. Oth-

erwise, the second smallest eigenvalue will be zero.

To ensure that the component being partitioned is connected, a preprocessing stage precedes the spectral analysis. In this stage, the links with zero flow are identified. If those links separate the network into components such that each component has positive intra-flow, the spectral partitioning is performed for each component separately. However, in transportation networks, it is more likely to observe multiple components where only one component has flow. In our analysis, this occurred due to the existence of peripheral links that do not have any flow but are included in the network geometry. In this case, those links are ignored since they are not used, and should not influence the partitioning of the main component.

Another consideration is the availability of flow values for the links in the network to perform initial partitioning. An approximate link flow solution could be obtained by solving static traffic assignment to a high gap value. Alternatively, we can heuristically approximate flows across the network based on past sensor measurements.

Algorithm 3 Flow Weighted Spectral Partitioning

Pre-process the network to remove links with zero flow:
if removing zero flow links creates multiple components with positive flow **then** partition each component separately

Flow-weighted spectral partitioning:
(1) Calculate the flow weighted graph Laplacian
(2) Normalize the graph Laplacian
(3) Get the eigenvector to be used for partitioning
(4) Order the nodes of the graph based on the eigenvector
(5) Partition the network by dividing the ordered node list based on the sign of the corresponding eigenvector elements

Determine if the obtained partitions should be divided further based on number of drones available:
if further partitioning is needed **then**
repeat flow-weighted spectral partitioning for each subnetwork

4.3 Shortest Domain Decomposition Algorithm

To evaluate the performance of the flow-weighted spectral partitioning algorithm, we implement an agglomerative heuristic that was developed by Johnson et al. [36]. This heuristic is referred to as the shortest domain decomposition algorithm (SDDA). The algorithm works in an agglomerative fashion and constructs a given number of partitions such that the number of boundary nodes between the subnetworks is minimized and the partitions are balanced in size. SDDA only depends on the topological properties of the graph. This feature is desirable when limited information is available on link costs, flow between origins and destinations, or other data that could form the basis of a partitioning algorithm.

The sequential steps of the algorithm are shown in Algorithm 4. The algorithm constructs the partitions by identifying source nodes which are "far" from each other given a distance measure. Specifically, the number of links on a breadth-first search tree between two nodes is used as the distance measure. This measure indicates the extent of separation of two nodes and is used to determine association of a node to the source nodes of the partitions. The reader is referred to Johnson et al. [36] for more information on this algorithm.

Algorithm 4 Shortest domain decomposition algorithm [36]

Initialization:

Let n_s be the number of subnetworks/partitions to be generated

(1) Set $R_s^n :=$ MAX

Determining first source node:

(2) Set the rank of each node as the sum of the number of incoming and outgoing links

(3) Choose the node with lowest rank s_1 as the first source node

Updating the rank and determining other source nodes:

for i in $2 : n_s$ **do**

(4) Perform breadth-first search from every source node, $s_j \quad \forall 1 \leq j < i$

(5) Determine the rank of node n as a $(i-1)$-dimensional vector whose elements are the distance of node n from source nodes s_j where $1 \leq j < i$

(6) Choose the node which has the highest total rank (sum of all elements in the rank vector)

(7) Resolve ties in favor of nodes which have minimum value of the sum of pair-wise difference between each element of the rank vector

(8) Assign the chosen node as the i-th source node s_i

Populate subdomain associated with each source node:

(9) For each node, assign it to the source node to which it has the minimum distance

Identify system boundary nodes and allocate the subnetworks:

for $(i, j) \in A$ **do**

if i and j are assigned to different source nodes **then**

(10) Add i and j to the set of boundary nodes.

4.4 Network Partitioning Results

We compare the performance of Algorithms 4 and 5 on three standard test networks: Anaheim, Austin, and Chicago sketch [59]. To determine link flow values for the flow-weighted spectral partitioning algorithm, we run static traffic assignment on the test networks (demand values, steady state link costs, and network features are provided in [59]). We note that computation time needed for partitioning is insignificant for both SDDA and flow-weighted spectral partitioning (less than 1 second on a 3.3 GHz machine with 8 GB RAM), and is thus not considered in our analysis.

Table 4.1 shows the resulting inter-flow when the algorithms are applied on the test networks. Unless mentioned otherwise, we generate two subnetworks from each network. The results show that in all cases, except the Chicago sketch network, the spectral partitioning algorithm generates subnetworks with lower inter-flow. This result is expected since the objective of the flow-weighted spectral partitioning method is to minimize the balanced inter-flow while SDDA does not consider network flows.

Table 4.1: Comparison of network partitioning algorithms

Network	**Inter-flow**
Austin (SDDA)	186161
Austin (Spectral)	137940
Austin (4 subnets, SDDA)	368718
Austin (4 subnets, spectral)	296870
Anaheim (SDDA)	81991
Anaheim (Spectral)	56539
Chicago sketch (SDDA)	154791
Chicago sketch (Spectral)	201603

We plot the partitions for the Austin network divided into four subnetworks using Algorithms 4 and 5. The results are shown in Figure 4.1. From this figure, we observe that SDDA generates peripheral partitions (partition 3). Deploying a drone to monitor solely this portion of the network is undesirable since the computational cost will not be properly divided across drones.

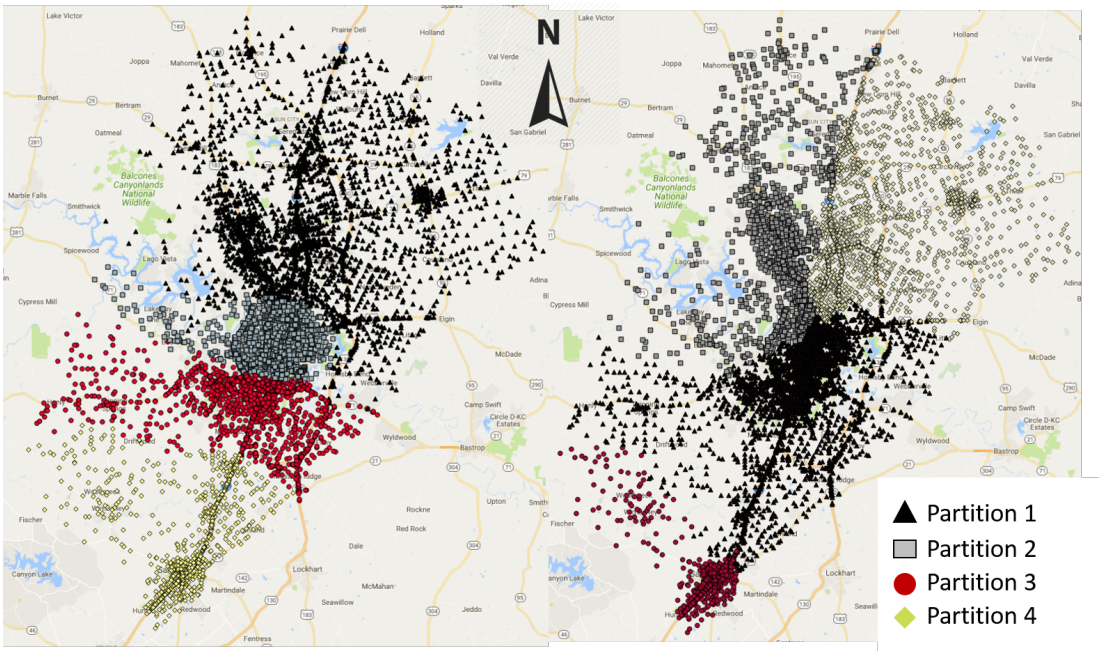

Figure 4.1: Partitioning of Austin regional network into four partitions. Left: flow-weighted spectral partitioning. Right: SDDA.

The Chicago sketch network partitions have lower inter-flow when generated using the SDDA algorithm. However, after further analysis, we observe that the SDDA partitions are heavily imbalanced with one subnetwork containing 90% of the flow. The spectral partitioning flow balance requirement will prohibit such partitions. In terms of drone deployment, if we assign drones to SDDA subnetworks and the geographic footprint of the subnetworks is comparable, then this would imply that we are focusing drone resources on uncongested areas with low impact on total delays. On the other hand, if the flows in each subnetwork are heavily imbalanced and one of the subnetworks is geographically a small portion of the full network, then the partitioning is influenced by level of detail and we are assigning drone resources inappropriately.

4.5 Conclusions

In this chapter, we proposed a flow-weighted spectral partitioning algorithm that divides a network into subnetworks monitored by separate drones. The algorithm is designed to minimize the inter-flow between subnetworks and to maintain subnetworks that are balanced by flow. These objectives ensure that (1) estimates in one subnetwork are not influenced by the quality of estimates in adjacent subnetworks, (2) the computational cost is divided among drones, and (3) drones are abundantly deployed in congested portions of the network where accurate traffic state and incident severity estimates are needed for congestion management strategies. We show that the proposed flow-weighted spectral partitioning algorithm is superior to an existing agglomerative heuristic.

Chapter 5

Conclusions & Future Work

Non-recurrent congestion is a primary source of travel time variability and congestion delays. Traditional data-driven methods for detecting non-recurrent congestion are not capable of simultaneously estimating traffic conditions and are susceptible to false alarms. On the other hand, methods that simultaneously estimate traffic conditions and incident severity suffer in congested conditions where it is difficult to distinguish between observations from congestion under normal operating conditions and similar observations that result due to reductions in capacity.

In this thesis, we propose a coupled planning and estimation framework that relies on unmanned aerial vehicles (drones) to generate observations which minimize the uncertainty on traffic states and certain parameters that reflect incident severity. Specifically, we develop an online one step lookahead algorithm that uses a dual state ensemble Kalman filter (EnKF) to determine the uncertainty minimizing drone path at every time step. The algorithm first computes the future state error covariance matrices based on expected observations via the dual state EnKF. Then, the drone moves in the direction of paths with minimal future variance on traffic state and model parameters. After the drone moves, it generates density and incident measurements. This data is used with ground sensor measurements to update the state error covariance matrices using a global dual state EnKF. Given the new drone location and updated state error covariance matrices, we repeat the

process to determine the next drone move.

We test the planning-estimation framework on a freeway network and compare its performance against a dual EnKF estimation procedure that is not aided by drone observations. The drone assisted dual EnKF shows significant improvement in estimation capabilities under congested conditions. In particular, the drone observations allow us to determine the exact road condition up to observation error in congested conditions. However, the dual state EnKF without a drone is not capable of estimating incident parameters in congested conditions due to the inability to distinguish between traffic patterns from capacity reductions and similar patterns from shock waves in incident-free conditions.

To plan for multiple drones working in parallel, we investigate partitioning algorithms that divide the network into subnetworks monitored by separate drones. The partitioning objectives we seek are (1) minimize the inter-flow between subnetworks and (2) maintain balanced subnetworks. The inter-flow minimization objective aims to reduce the dependence of estimates in a subnetwork on estimates in adjacent subnetworks. Maintaining balanced subnetworks divides the computational cost across drones. We ensure that drones are properly deployed in congested portions of the network by balancing the subnetworks by total flow. We implement a flow-weighted spectral partitioning algorithm that generates subnetworks according to the desired objectives. We also show that the proposed algorithm is superior to an existing agglomerative heuristic that does not consider traffic flows.

5.1 Summary of Contributions

In brief, the contributions of this thesis are as follows:

1. We analyze the variation in traffic model parameters subject to incidents such as work zones, roadside accidents, and adverse weather conditions.

2. We develop a dual state EnKF algorithm for simultaneous estimation and incident detection.

3. We develop a coupled estimation and planning framework that routes a drone towards road segments with high uncertainty on parameters and states. While detecting incidents in congested conditions is not possible using ground sensor data only, a drone enables incident detection via its image processing capabilities.

4. We implement a flow-weighted spectral partitioning algorithm that generates flow balanced partitions with minimal inter-flow. The resulting subnetworks can be used to effectively distribute multiple drones for monitoring large networks.

5.2 Future Work

Applying the proposed estimation and routing framework on larger networks raises computational challenges pertaining to the number of candidate paths that the drone can follow at every time step. In future work, we aim to improve the scalability of the framework by developing alternative methods for analyzing candidate paths. Specifically, when paths share several road segments, the expected

future observations and corresponding uncertainties that result from traversing these paths will be similar. We can use this similarity to limit the number of candidate paths evaluated.

In addition, we will investigate methods for operating multiple drones working in parallel. While the proposed network partitioning method provides guidelines for deploying multiple drones working in parallel, the partitioning does not account for the possibility of sharing information between drones. In case of information sharing, we need to develop novel deployment and path planning strategies that exploit this capability.

We will also investigate the applicability of the estimation and planning framework in different transportation settings. For example, drones have been recently used for parcel delivery. When a drone is delivering parcels in tandem with a truck, the drone can provide traffic state estimates that inform the truck about the surrounding congestion state. These state estimates can be used to route the truck-drone system towards least cost paths that are subject to parcel delivery constraints.

Bibliography

[1] K. Kanistras, G. Martins, M. J. Rutherford, and K. P. Valavanis, "Survey of unmanned aerial vehicles (UAVs) for traffic monitoring," Springer, 2015.

[2] T. Seo, A. M. Bayen, T. Kusakabe, and Y. Asakura, "Traffic state estimation on highway: A comprehensive survey," *Annual Reviews in Control*, vol. 43, 2017.

[3] R. Wang, D. B. Work, and R. Sowers, "Multiple model particle filter for traffic estimation and incident detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, 2016.

[4] R. Wang and D. B. Work, "Interactive multiple model ensemble Kalman filter for traffic estimation and incident detection," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2014.

[5] A. Willsky, E. Chow, S. Gershwin, C. Greene, P. Houpt, and A. Kurkjian, "Dynamic model-based techniques for the detection of incidents on freeways," *IEEE Transactions on Automatic Control*, vol. 25, no. 3, 1980.

[6] R. Wang, S. Fan, and D. B. Work, "Efficient multiple model particle filtering for joint traffic state estimation and incident detection," *Transportation Research Part C: Emerging Technologies*, vol. 71, 2016.

[7] A. Dabiri and B. Kulcsár, "Freeway traffic incident reconstruction – a bi-parameter approach," *Transportation Research Part C: Emerging Technologies*, vol. 58, 2015.

[8] Y. Wang, M. Papageorgiou, M. Messmer, P. Coppola, A. Tzimitsi, and A. Nuz-

zolo, "An adaptive freeway traffic state estimator," *Automatica*, vol. 45, no. 1, 2009.

[9] K. K. Sanwal, K. Petty, J. Walrand, and Y. Fawaz, "An extended macroscopic model for traffic flow," *Transportation Research Part B: Methodological*, vol. 30, no. 1, 1996.

[10] S. Blandin, A. Couque, A. Bayen, and D. Work, "On sequential data assimilation for scalar macroscopic traffic flow models," *Physica D: Nonlinear Phenomena*, vol. 241, no. 17, 2012.

[11] A. Skabardonis, P. Varaiya, and K. Petty, "Measuring recurrent and nonrecurrent traffic congestion," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1856, 2003.

[12] B. Anbaroglu, B. Heydecker, and T. Cheng, "Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks," *Transportation Research Part C: Emerging Technologies*, vol. 48, 2014.

[13] Y. J. Stephanedes and A. P. Chassiakos, "Freeway incident detection through filtering," *Transportation Research Part C: Emerging Technologies*, vol. 1, no. 3, 1993.

[14] Y. J. Stephanedes, A. P. Chassiakos, and P. G. Michalopoulos, "Comparative performance evaluation of incident detection algorithms," *Transportation Research Record*, vol. 1360, 1992.

[15] E. Parkany and C. Xie, "A review of incident detection technologies, algorithms and their deployments: What works and what doesn't," Tech. Rep. NETCR37, University of Massachusetts Transportation Center, 2005.

[16] H. J. Payne and S. C. Tignor, "Freeway incident-detection algorithms based on decision trees with states," *Transportation Research Record*, 1978.

[17] H. Dia and G. Rose, "Development and evaluation of neural network freeway incident detection models using field data," *Transportation Research Part C: Emerging Technologies*, vol. 5, no. 5, 1997.

[18] Y. J. Stephanedes and X. Liu, "Artificial neural networks for freeway incident detection," *Transportation Research Record*, 1995.

[19] R. L. Cheu and S. G. Ritchie, "Automated detection of lane-blocking freeway incidents using artificial neural networks," *Transportation Research Part C: Emerging Technologies*, vol. 3, no. 6, 1995.

[20] H. Teng and Y. Qi, "Application of wavelet technique to freeway incident detection," *Transportation Research Part C: Emerging Technologies*, vol. 11, 2003.

[21] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, 1960.

[22] G. Evensen, "The ensemble Kalman filter: Theoretical formulation and practical implementation," *Ocean Dynamics*, vol. 53, no. 4, 2003.

[23] S. Haykin, "Kalman filters," in *Kalman Filtering and Neural Networks*, John Wiley & Sons, 2001.

[24] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068, International Society for Optics and Photonics, 1997.

[25] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, 2000.

[26] A. Doucet, N. De Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods," in *Sequential Monte Carlo Methods in Practice*, Springer, 2001.

[27] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, 2000.

[28] Y. Wang and M. Papageorgiou, "Real-time freeway traffic state estimation based on extended Kalman filter: A general approach," *Transportation Research Part B: Methodological*, vol. 39, no. 2, 2005.

[29] Z. Sun, W. Jin, and S. G. Ritchie, "Simultaneous estimation of states and parameters in Newell's simplified kinematic wave model with Eulerian and Lagrangian traffic data," *Transportation Research Part B: Methodological*, vol. 104, 2017.

[30] J. Y. J. Chow, "Dynamic UAV-based traffic monitoring under uncertainty as a stochastic arc-inventory routing policy," *International Journal of Transportation Science and Technology*, vol. 5, no. 3, 2016.

[31] H. Ikeda, Y. Kaneko, T. Matsuo, and K. Tsuji, "Abnormal incident detection system employing image processing technology," in *Proceedings 199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, 1999.

[32] K. Wang, X. Jia, and S. Tang, "A survey of vision-based automatic incident detection technology," in *IEEE International Conference on Vehicular Electronics and Safety, 2005*, 2005.

[33] H. Etemadnia, K. Abdelghany, and A. Hassan, "A network partitioning methodology for distributed traffic management applications," *Transportmetrica A: Transport Science*, vol. 10, no. 6, 2014.

[34] N. Garg, V. V. Vazirani, and M. Yannakakis, "Approximate max-flow min-(multi) cut theorems and their applications," *SIAM Journal on Computing*, vol. 25, no. 2, 1996.

[35] M. Saeedmanesh and N. Geroliminis, "Clustering of heterogeneous networks with directional flows based on "snake" similarities," *Transportation Research Part B: Methodological*, vol. 91, 2016.

[36] P. Johnson, D. Nguyen, and M. Ng, "Large-scale network partitioning for decentralized traffic management and other transportation applications," *Journal of Intelligent Transportation Systems*, vol. 20, no. 5, 2016.

[37] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, 1998.

[38] D. A. Spielman, "Spectral graph theory and its applications," in *48th Annual IEEE Symposium on Foundations of Computer Science*, IEEE, 2007.

[39] D. A. Spielman and S. Teng, "Spectral partitioning works: Planar graphs and finite element meshes," *Linear Algebra and its Applications*, vol. 421, no. 2-3, 2007.

[40] M. E. J. Newman, "Spectral methods for community detection and graph partitioning," *Physical Review E*, vol. 88, no. 4, 2013.

[41] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, 2007.

[42] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, 2000.

[43] M. G. Bell, F. Kurauchi, S. Perera, and W. Wong, "Investigating transport network vulnerability by capacity weighted spectral analysis," *Transportation Research Part B: Methodological*, vol. 99, 2017.

[44] S. Martinez, G. Chatterji, D. Sun, and A. M. Bayen, "A weighted-graph approach for dynamic airspace configuration," in *Proceedings of the AIAA Conference on Guidance, Navigation, and Control (GNC). American Institute of Aeronautics and Astronautics*, 2007.

[45] Y. Ma, Y. Chiu, and X. Yang, "Urban traffic signal control network automatic partitioning using Laplacian eigenvectors," in *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference On Intelligent Transportation Systems*, IEEE, 2009.

[46] C. F. Daganzo, "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transportation Research Part B: Methodological*, vol. 28, no. 4, 1994.

[47] C. F. Daganzo, "The cell transmission model, Part II: Network traffic," *Transportation Research Part B: Methodological*, vol. 29, no. 2, 1995.

[48] S. Godunov, "A difference method for the numerical calculation of discontinuous solutions of hydrodynamic equations," *Mathematics Sbornik*, vol. 47, no. 3, 1959.

[49] P. T. V. Group, "PTV Vissim." http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/, 2018.

[50] G. Evensen, *Data Assimilation: The Ensemble Kalman Filter*. Springer Science & Business Media, 2009.

[51] E. S. Canepa and C. G. Claudel, "Networked traffic state estimation involving mixed fixed-mobile sensor data using Hamilton-Jacobi equations," *Transportation Research Part B: Methodological*, vol. 104, 2017.

[52] S. Haykin, *Kalman Filtering and Neural Networks*, vol. 47. John Wiley & Sons, 2004.

[53] J. W. C. Van Lint, S. P. Hoogendoorn, and A. Hegyi, "Dual EKF state and parameter estimation in multi-class first-order traffic flow models," in *IFAC Proceedings Volumes*, vol. 41, 2008.

[54] A. Hegyi, D. Girimonte, R. Babuska, and B. De Schutter, "A comparison of filter configurations for freeway traffic state estimation," in *Intelligent Transportation Systems Conference, 2006. ITSC'06*, IEEE, 2006.

[55] D. Ucinski, *Optimal Measurement Methods for Distributed Parameter System Identification*. CRC Press, 2004.

[56] A. Atkinson, A. Donev, and R. Tobias, *Optimum Experimental Designs, with SAS*, vol. 34. Oxford University Press, 2007.

[57] R. Sim and N. Roy, "Global A-optimal robot exploration in SLAM," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automatio*, IEEE, 2005.

[58] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1. Athena Scientific, 4th ed., 2017.

[59] B. Stabler, "Transportation networks for research." https://github.com/bstabler/TransportationNetworks, 2017.