Real-Time Calibration of Large-Scale Traffic Simulators: Achieving Efficiency Through the Use of Analytical Models

by

Kevin Zhang

B.A., Yale University (2012)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2020

Author .....
Sloan School of Management
July 30, 2020

Certified by .....
Carolina Osorio
Associate Professor
Thesis Supervisor

Accepted by .....
Georgia Perakis
William F. Pounds Professor of Management Science
Co-Director, Operations Research Center

Real-Time Calibration of Large-Scale Traffic Simulators: Achieving Efficiency Through the Use of Analytical Models

by

Kevin Zhang

Submitted to the Sloan School of Management
on July 30, 2020, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

Stochastic traffic simulators are widely used in the transportation community to model real-world urban road networks in applications ranging from real-time congestion routing and control to traffic state prediction. Online calibration of these simulators plays a crucial role in achieving high accuracy in the replication and prediction of streaming traffic data (i.e., link flows, densities). In order to be relevant in a real-time context, the problem must also be solved within a strict computational budget. The primary goal of this thesis is to develop an algorithm that adequately solves the online calibration problem for high-dimensional cases and on large-scale networks.

In the first half, a new online calibration algorithm is proposed that incorporates structural information from an analytical metamodel into a general-purpose extended Kalman filter framework. The metamodel is built around a macroscopic network model that relates calibration parameters to field measurements in an analytical, computationally tractable, and differentiable way. Using the metamodel as an analytical approximation of the traffic simulator improves the computational efficiency of the linearization step of the extended Kalman filter, making it suitable for use in large-scale calibration problems. The embedded analytical network model provides a secondary benefit of making the algorithm more robust to simulator stochasticity compared with traditional black-box calibration methods.

In the second half, the proposed algorithm is adapted for the case study of online calibration of travel demand as defined by a set of time-dependent origin-destination matrices. First, an analytical network model relating origin-destination demand to link measurements is formulated and validated on the Singapore expressway network. Next, the proposed algorithm is validated on a synthetic toy network, where its flexibility in calibrating to multiple sources of field data is demonstrated. The empirical results show marked improvement over the baseline of offline calibration and comparable performance to multiple benchmark algorithms from the literature. Finally, the proposed algorithm is applied to a problem of dimension 4,050 on the Singapore expressway network to evaluate its feasibility for large-scale problems. Empirical re-

sults confirm the real-time performance of the algorithm in a real-world setting, with strong accuracy in the estimation of sensor counts.

Thesis Supervisor: Carolina Osorio
Title: Associate Professor

Acknowledgments

First and foremost, I would like to thank my advisor, Professor Carolina Osorio, who has been a constant source of insight and support through my time as a PhD student at MIT. Her research instincts and dedication to her students have been inspiring to witness, and her encouragement and guidance in both academic research and life in general have been invaluable to me.

I would like to thank Dr. Ravi Seshadri, who has been an incredible collaborator in my PhD research. For the last several years, he has provided helpful feedback every step of the way and been a great resource in working with DynaMIT. I also owe much gratitude to my committee members, Professor Saurabh Amin and Professor Hamsa Balakrishnan, for their advice on how to shape the arguments of this thesis and their feedback throughout the process. Their words have meant a lot.

Many others have provided valuable input during my PhD, particularly fellow members of the Osorio Lab. Linsen, Chao, Evan, Jing, Tianli, Timothy, and Nate—thank you for your many suggestions, answers, and encouragements since the very beginning. I must also thank Haizheng Zhang, other members of the ITS Lab at MIT, and researchers at SMART for support with DynaMIT and development of the Singapore network case study. Finally, a huge thanks goes to all of the administrative staff in both the Operations Research Center and the CEE Department, especially Laura Rose, Andrew Carvalho, and Roberta Pizzinato. Their competence and patience have made this experience infinitely more manageable and enjoyable.

I could not have done this without the support of my friends and family. I am fortunate to have many people in my life cheering me on and willing to offer a hand no matter the reason, and I appreciate each and every one of you. To those of you who have overlapped in Boston with me—Julia, Christine, Devon, Diana, Naomi, Emily, Chidi, Daisy, Lauren—thank you for the endless meals, movies, walks, and drinks that have filled the last six years. To those farther away, especially Katie and Kelly, your conversations, gifts, and visits have been a great source of joy. And finally to my parents and my sister Katie, thank you for everything.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Traffic congestion on urban road networks is an age-old problem that has only become more important with increased urbanization. According to the World Bank, the percentage of the entire population of the United States living in urban areas has steadily increased from 73.6% in 1970 to 82.3% in 2018 [57]. Similar statistics hold globally; the United Nations projects that 68% of the world's population is expected to live in urban areas by 2050 compared to 55% in 2018 [61]. The impacts of increased urbanization and the associated traffic congestion are manifold, ranging from increased energy consumption and adverse environmental effects like air pollution to lost productivity. As a result, there is a growing need for better traffic management of existing capacity on urban road networks. The traffic management centers responsible for this guidance use traffic simulators to make decisions about traffic control and congestion mitigation, often in a real-time setting.

Traffic simulators are widely used in the transportation community to model real-world traffic networks and to estimate and predict traffic conditions from data received through intelligent transportation systems (ITS). Applications of these simulators range from real-time congestion routing and control to offline scenario planning. Traffic simulators are comprised of supply and demand models that inform driver behavior, network behavior, and their interactions within the simulator with the goal of replicating traffic dynamics observed in the real-world network. These models can be at various levels of detail, from microscopic models that model individual drivers and

their travel behavior decisions to macroscopic models that model higher-level traffic dynamics. These models can also be analytical or simulation-based. Additionally, as the interactions between demand and supply grow more complicated with new technological advancements in vehicle-to-vehicle and vehicle-to-infrastructure communications, the models that make up these traffic simulators have also grown more intricate.

The other growing trend in transportation is an explosion of data available to traffic management centers. As data collection tools for transportation networks have developed and become more prevalent, methods driven by the new wealth of data available have been the focus of much research in transportation and traffic network modeling. The available data, specifically link- and route-specific information like flows, travel times, and speeds collected by induction loop detectors, probe vehicles, and increasingly mobile phone data, has led researchers to develop methods for incorporating new sources of information into existing transportation solutions, in service of estimation of traffic behavior and improvement of forecasting power. Models for real-time traffic estimation and prediction give managers of traffic networks crucial data to make informed decisions about policies to reduce congestion and travel times.

These data-driven methods are primarily concerned with achieving high accuracy in replication and prediction of real-world traffic information like flows, travel times, and speeds. Since traffic networks can encompass hundreds or thousands of links and intersections, the scalability and computational efficiency of a data-driven method are also important considerations. With the increase in both intricacy of traffic simulator models and incorporation of newly available data, there is a critical need for methods guiding the operations of these urban road networks that are scalable, reliable, and computationally efficient, particularly in the context of real-time traffic management. The use of traffic simulators by practitioners for real-time responsiveness will only become more imperative as the effects of growing urbanization and congestion are felt worldwide.

The focus of this thesis is the online calibration of stochastic simulation-based, rather than analytical or deterministic, traffic models. These traffic simulators are

comprised of simulation-based models for demand and supply and cannot be represented by an analytical or closed-form expression. In general, the online calibration problem aims to identify model parameter inputs for every discrete time interval in the simulation period that lead to network state outputs (e.g., sensor counts, link travel times) from the simulator consistent with real-time field measurements observed on the real-world road network. The aim of this thesis is to develop an online calibration algorithm that solves the online calibration problem for high-dimensional cases (i.e., thousands of model parameters) and on large-scale networks to a high degree of accuracy.

In addition to the goal of accuracy, several other factors play a significant role in online calibration—algorithms for online calibration must be computationally efficient and able to find parameter values in a real-time setting, often on the order of 5 to 30 minutes, to be of practical value, and algorithms should be robust to simulator stochasticity. As seen in Section 1.3, many online calibration algorithms from the literature treat the simulator as a black box, only utilizing it for basic inputs and outputs. While this approach is model-agnostic and thus can be applied across problem settings, algorithms of this type often rely on computationally costly estimation methods, limiting their scalability to larger networks. The primary goal of this thesis is to embed problem-specific structural information within these existing algorithms to improve their computational efficiency for large-scale networks. This structural information is leveraged through the use of analytical traffic models, which are combined with the black-box algorithm. This leads to the following research question: can an online calibration framework be developed that combines existing online calibration algorithms with analytical network model information in a way that leads to good calibration performance within a tight computational budget for large-scale real-world networks?

In the design of this new framework, it should have the following qualities, which are addressed throughout the thesis both in the algorithm development and in the empirical case studies:

1. the algorithm should be general-purpose, in that it is able to jointly calibrate

parameters across several types of models,

2. the algorithm should be able to incorporate different types of field measurement data,

3. the algorithm should be robust to the stochasticity of the traffic simulator,

4. the algorithm should be of practical use to transportation practitioners, i.e., it should be able to solve high-dimensional online calibration problems on large-scale networks with a limited computational budget.

1.1 Dynamic Traffic Assignment systems

While traffic simulators range in specificity from microscopic simulators that model distinct agents and their individual movements to macroscopic simulators that model aggregate behavior, all stochastic traffic simulators operate under a framework in which driver behavior responds iteratively to network conditions and the actions of other drivers. The intricacy of the interactions being modeled in these transportation systems leads to the use of what are called Dynamic Traffic Assignment (DTA) systems [16, 40].

Figure 1-1 shows a general schematic of a Dynamic Traffic Assignment framework. As seen in the figure, DTA systems are comprised of supply and demand models that inform driver and network behavior and the supply-demand interactions between the two within the simulator. Demand models, represented by the box on the left, include demand generation, often through time-dependent origin-destination (O-D) matrices, and route choice. In addition, demand models may include information on driver classes, travel mode preference, and both access and response to guidance information. Supply models, represented by the box on the right, dictate link attributes and movement between links of the road network. These models specify supply behavior such as speed-density relationships on network links, lane connections, and link output capacities based on network topology and traffic signalization. Models of how traffic incidents and other events occur and propagate through the network are also

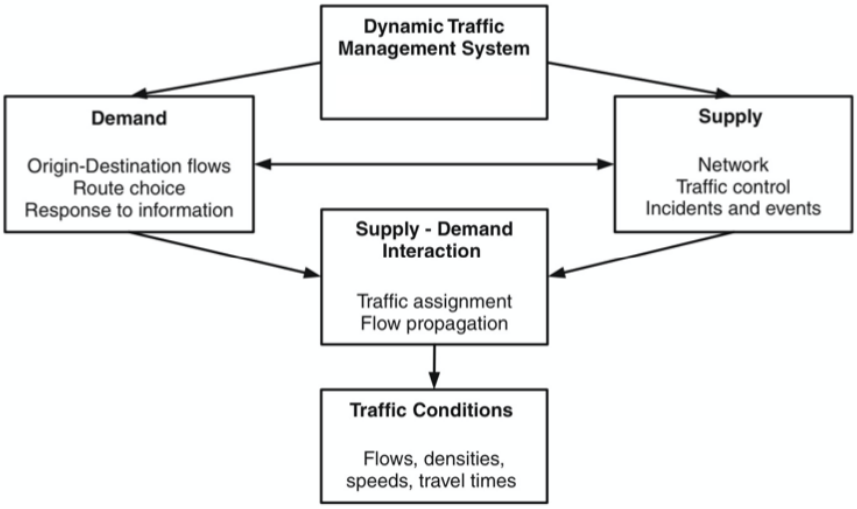

Figure 1-1: General DTA framework adapted from [15]

included in the supply models. Through the simulation-based interaction between these supply and demand models, often an iterative process, the DTA system models traffic conditions on the network through the traffic assignment of drivers and flow propagation according to supply models. The result of these simulations is a detailed picture of the road network and its traffic conditions through time, and from this, key network attributes like link flows, densities, speeds, and travel times can be observed from the DTA system.

Simulation-based Dynamic Traffic Assignment systems can be used in both planning and real-time contexts. Real-time capabilities, which are the focus of online calibration, are comprised of two functions, state estimation and prediction-based information generation. The goal of state estimation is to infer current traffic conditions on the network as represented by information like O-D demand, link flows, queue lengths, velocities, and densities at various locations on the network. Real-time information from traffic sensors is combined with offline historical data to estimate the state, which also relies on driver behavioral models and guidance information. The goal of the prediction functions is to estimate the state of the network in the near future and use that to create information provided to travelers and control strategies. Prediction-based information that is generated can consist of pre-trip demand levels,

O-D flow prediction, network state prediction, and even guidance generation.

1.2 Parameter calibration problem definition

The Dynamic Traffic Assignment system models contain a large number of parameters and inputs that must be tuned to ensure realistic and reliable output from the simulator, and accurate calibration of these model parameters is a fundamental step in using stochastic traffic simulators to reproduce the behavior of real-world traffic systems. During the calibration process, estimates of the model parameters and inputs are calculated based on field data from the road network. Calibration generally occurs in two regimes—offline calibration and online calibration.

Offline calibration provides a baseline estimate of model parameters, usually derived from a set of historical field measurements. The goal of offline calibration is to find settings for supply and demand parameters that correspond to typical traffic dynamics on a network and average network conditions over the period represented by the historical data. Several sets of offline calibrated parameters values may be computed for different network settings, like time of day, day of week, or season, but real-time field data is not utilized during offline calibration. Algorithms for offline calibration identify good quality parameter estimates, which can be re-updated as new field observations are added to a database of historical information.

Online calibration is the process of tuning simulator parameters to account for fluctuations in traffic measurements in a real-time setting. Given a baseline set of historical or long-run average parameter values from offline calibration, the online calibration problem is focused on adjusting those parameter values in response to streaming field data as they are being observed on the real-world road network. By incorporating field data reflecting current network conditions, using model parameters obtained during online calibration allows traffic management centers to use stochastic traffic simulators for real-time applications. For these applications, such as real-time congestion routing and traffic control for incident management, the accuracy and specificity of the online calibration algorithm is important, as the goal

is to model time-dependent phenomena like rush hour demand and weather-related network constraints and to reflect prevailing traffic conditions in order to provide context-specific decisions. Accurate online calibration is also necessary for effective prediction of future traffic conditions that feed into real-time guidance decisions. Because these applications must operate in a real-time context, runtime considerations for online calibration algorithms are also important. Online calibration algorithms need to provide solutions in a fixed time window, usually on the order of 5 to 30 minutes, to retain their relevancy; thus, they must be computationally inexpensive and efficient. For large-scale networks that have thousands of model parameters to calibrate, just simulating traffic conditions over a 5-minute time frame using a stochastic traffic simulator can take a couple of minutes to run depending on the granularity of the simulator models. In these situations, the computational efficiency of the online calibration algorithm becomes of the utmost importance.

Within the broader parameter calibration problem, a crucial and well-studied subproblem is real-time origin-destination (O-D) demand calibration of stochastic traffic simulators. Time-dependent O-D demand matrices are a significant input for the demand models of traffic simulators and are a key component for online calibration. The O-D calibration problem must combine historical and real-time data to obtain dynamic (time-dependent) demand matrices, which are then used to seed drivers, their trips, and their route choice behavior in the simulator. Additionally, online O-D calibration can then feed into prediction of O-D matrices for future time periods that significantly impact the ability of practitioners to anticipate and react to future traffic dynamics.

1.3 Literature review

The online calibration problem has received significant attention from the transportation research community, especially with the growing use of DTA systems, the increasing intricacy of tools for modeling traffic behavior on road networks, and richer sources of traffic data. While this thesis focuses on online calibration of stochastic

simulation-based traffic models, the literature review surveys research across several types of models, which can generally be split between stochastic and deterministic simulators, and between simulation-based models and analytical models. Depending on the type of models being calibrated, different methodologies have been proposed for tackling the online calibration problem.

In general, there are two popular modes for approaching the problem. The first, motivated by applications with simulation-based models and stochastic traffic simulators, is a more black-box approach that does not make any assumptions on the specific models that comprise the simulator. The aim of this family of general-purpose methods is to be suitable for the calibration of any combination of model parameters using any type of data available from the field. In particular, parameters for demand and supply models may be calibrated jointly. The second mode of approach is motivated more by a specific problem context, and attempts to utilize model-based or analytical information to solve an online calibration problem. By focusing on a particular set of model parameters (i.e., speed-density parameters, car-following model parameters, or demand matrices) or type of field data (i.e., sensor counts or probe vehicle travel times), these methods aim to exploit structural information to efficiently solve the calibration problem. The majority of these problem-specific approaches tackle the calibration of dynamic origin-destination (O-D) demand parameters, which is also the focus on the case studies in Chapters 4 and 5 of this thesis.

Regardless of the mode of approach, all online calibration algorithms hope to provide accurate parameter tuning in a real-time setting. Recent work has also focused on the scalability of these approaches for use on high-dimensional calibration problems that are practically relevant for transportation practitioners. For a sense of the scope of these online calibration problems, some of the largest road networks used as case studies among research cited in this literature review are comprised of thousands of links and model freeway and arterial networks in major world cities like Stockholm [6], San Francisco [33], and Singapore [52]. The number of parameters calibrated in each of these case studies is 462, 769, and 4121, respectively. In general, the number of parameters being calibrated can easily reach an order of magnitude in

the hundreds or thousands. In addition, to be considered a real-time approach, the online calibration algorithm should identify parameter estimates within a limited time budget, typically depicted as the time between consecutive measurements from the field. For the case studies mentioned, the time windows range from 5 to 15 minutes. A modified version of the Singapore road network, used in case studies by Prakash et al. [52] and Zhang [73], is used for the case study discussed in Chapter 5.

As mentioned, one methodology suggested for the online calibration problem is to create a solution framework that can be applied without any assumptions on the types of models used by the traffic simulator. The goal of these general-purpose frameworks is to be adaptable for any type of traffic model (i.e., analytical, simulation-based, microscopic, macroscopic, etc.), any set of calibration parameters across both demand and supply models, and any type of field data (i.e., link flows, route travel times, turning fractions, etc.). As real-time data collection tools for transportation networks have become more advanced, there is interest in algorithms that are flexible in incorporating data of many different types [13]. In addition, as argued by Antoniou [3], methods that are able to jointly calibrate model parameters provide an advantage over methods that calibrate only a subset of parameters and sequential calibration methods. For detailed DTA systems, the interactions among demand and supply models can be very intricate, and methods that model these interactions through the joint calibration of demand and supply parameters are shown to have improved empirical performance [3]. While these methods provide adaptability to various problem contexts, this generality also comes at the cost of computational efficiency and specificity. Model assumptions are intentionally not made, so the majority of general-purpose methods are black-box optimization methods that utilize simulator information only through simulator evaluations, with parameters as input and network measurements as output.

The vast majority of these general-purpose approaches to online calibration utilize a state-space model formulation of the problem, which provides a compact and scalable representation of the traffic state through time. An established methodology commonly found in the literature is to formulate the problem as a state-space

model with the parameters to be calibrated defined as the state [10, 75]. The model is then comprised of two types of equations—transition equations describe the evolution of the model parameters through time, while measurement equations relate the parameters to observable traffic data used in the calibration. The consequences of a state-space model formulation have been widely discussed in the literature. While the model provides a simple, and thus relatively compact, representation of the temporal evolution of model parameters, there is a trade-off between model simplicity, which impacts computational efficiency, and the fidelity of the model to real-world behaviors, especially as represented by the transition equations. Model inconsistencies, which measure the divergence from real-world network behavior, propagate within a state-space model framework and can significantly degrade predictive performance [36]. Modifications have been proposed to expand the size of the state to incorporate more network information [9] or to increase the degree of the autoregressive relationship modeled by the transition equation [75], but accommodations to improve model fidelity also increase the computational burden, especially for large-scale network applications.

The state-space model formulation is commonly solved using Kalman filtering techniques. Zhou and Mahmassani [75] calibrate O-D demand using a state-space model, assuming a stationary random process with constant mean and variance for the transition equation. The approach incorporates a polynomial trend component to model structural deviations in the parameters from the historical O-D demand pattern. While the Kalman filter solution algorithm is validated using data from a road network of freeways and main arterials in Irvine, California, it is not fully implemented within a real-time DTA system. Antoniou [3] develops a nonlinear state-space model defined by a state vector consisting of calibration parameters, an auto-regressive transition equation, and a nonlinear measurement equation given by the simulator being calibrated. Since the state-space model is nonlinear, the standard Kalman filter cannot be used and instead both an extended Kalman filter (EKF) and an unscented Kalman filter (UKF) method are proposed. Both solution methods are implemented in a case study on a freeway network in Southampton, UK, with a

problem dimension of 80 parameters. While both methods show good performance with average root-mean-square normalized (RMSN) errors below 20%, the extended Kalman filter approach is judged to be more straightforward in implementation.

Though the general-purpose framework developed by Antoniou was shown to have good performance in estimation and prediction of traffic state measurements, it is not practical for the real-time calibration of large-scale networks. Both of the proposed algorithms (EKF and UKF) incur a significant computational cost at each time step due to the number of simulator evaluations required to deal with the nonlinearity of the state-space model formulation. The extended Kalman filter approach approximates the gradient of the simulator at each time step, and since no analytical assumptions are made about the models being calibrated, the gradient estimation must rely on numerical methods. Antoniou suggests the use of central finite differences as a numerical gradient approximation method. For a calibration problem of dimension n, this requires $\mathcal{O}(n)$ simulator evaluations each time step, which is unwieldy for high-dimensional problems [5]. Additionally, numerical gradient estimation methods must contend with errors stemming from simulator stochasticity. For simulator evaluations at points differing by small perturbations, the random noise inherent in stochastic traffic simulator output may outweigh true measurement of the gradient. To counteract this stochasticity, multiple replications of the gradient estimate may be taken and averaged, which increases the computational burden of the gradient approximation step [73]. There is a trade-off between accuracy of the gradient approximation and the online computational cost of the algorithm. Similarly, the unscented Kalman filter runs a set of sample points (called "sigma points") through the simulator at each time step to approximate the first and second moments of the state vector distribution. The number of sigma points scales linearly with the dimension of the state vector. As these methods aim to jointly calibrate all model parameters, which for a large-scale problems may number in the thousands, these Kalman filtering algorithms cannot be implemented in real-time without significant modifications.

Since Antoniou's formulation of the online calibration problem as a nonlinear state-space model, many approaches have been proposed to improve the computa-

tional efficiency of the general-purpose Kalman filter approach for use in real time on problems of high dimension. Several focus on improving the gradient computation step of the extended Kalman filter. Antoniou proposed two variations of his algorithm in follow-up work—first, using simultaneous perturbation instead of finite differences to approximate the gradient [7]. The simultaneous perturbation method, as first proposed by Spall [56], only uses two random perturbations to approximate the gradient, significantly reducing online computational cost but with a noisier gradient estimate and a small deterioration in estimation performance, as shown by Antoniou in a case study on the I-405 freeway network in Irvine, California [7]. As an alternative, Antoniou proposes another algorithm in which a series of gradients are calculated offline and the average is used online; the method is called the "Limiting EKF" [3]. As shown in experiments calibrating traffic counts on a network representing the central district of Stockholm, the Limiting EKF algorithm is able to produce root-mean-square normalized errors in the range of 10% to 16% for estimation and prediction of field measurements for a real-world problem while significantly reducing the online computational cost compared to the finite differences approach [6]. Interestingly, Antoniou et al. also observe that the Limiting EKF shows robustness in the face of simulator noise compared to the finite differences approach due to averaging several noisy estimates of the gradient. Still, the fixed gradient estimate used by the Limiting EKF results in an added approximation and information loss; it does not model temporal aspects of the problem crucial for online calibration.

Other approaches improve the computational efficiency of the nonlinear Kalman filtering algorithm by reducing the dimension of the problem. Huang [34] and Zhang [73] propose a method for gradient estimation called partitioned simultaneous perturbation (PSP). Similarly, a dimensionality reduction approach using principal components analysis (PCA) is proposed by Djukic and integrated within the Kalman filtering framework [27]. Using a smaller number of principal components as the state reduces the dimension of the calibration problem while still capturing a large fraction of the variance of the system. A principal components analysis approach is also taken by Prakash [52]. In a case study on the Singapore expressway network,

4121 O-D demand parameters are calibrated using sensor counts. Using 80 principal components that explain 95% of the variance as the state vector, Prakash et al. are able to achieve better estimation and prediction results than the full-dimensional formulation while seeing a 50-fold reduction in computational cost. Finally, one more dimensionality reduction approach is proposed by Carrese et al. [20] based on the Local Ensemble Transformed Kalman Filter (LETKF) developed by Hunt et al. [35]. The LETKF algorithm transforms the problem from the space of state variables (i.e., calibration parameters) to a lower-dimensional space defined by the "ensemble" that parametrically models the distribution of the state variables. The method avoids a linearization step, similar to the unscented Kalman filter, but at a smaller and more scalable computational cost. Still, the simulator must be evaluated for each member of the ensemble, and the experiments cited only consider problems of dimension up to 36. Carrese et al. even acknowledge that for a large-scale network or a larger-sized ensemble, computational times may not be suitable for online application [20].

In contrast to online calibration approaches that construct a general-purpose framework that will work with any type of traffic models, other methods in the literature take a more narrow approach by focusing on a specific subset of parameters to calibrate or type of data from the field. While these methodologies are more restrictive in their application and make assumptions about the problem context or form of the models calibrated, they are also able to utilize problem-specific information through analytical models or in algorithm design. One such approach by Tavana and Mahmassani [59] focuses on the calibration of dynamic speed-density relations using transfer function methods, which capture temporal effects like serial correlation across the calibration period. Each transfer function is a bivariate time-series model for the speed-density relationship on a link, with density as the leading indicator and deviations of speeds from historical values as the dependent variable. The transfer function parameters are calibrated using link detector data. While a case study is run using data from a freeway network in San Antonio, the method is not incorporated within a simulation-based framework. Huynh et al. [36] extend the work by Tavana and Mahmassani by implementing the methodology in a real-time context,

with an adaptive least-squares optimization procedure for re-fitting speed-density parameters that allows for integration with a DTA system. Specifically, they apply the transfer function models for dynamic speed-density relations within DYNASMART, a simulation-based DTA system, in order to improve estimation of link speeds. An experiment is run on a network of the I-35 freeway with neighboring arterials and local streets as a proof-of-concept with promising results, though only one link is calibrated with the proposed method and no real traffic data is used. Finally, Qin et al. [53] extend the method even further by evaluating performance using real sensor data from an Irvine, California, network. While the dynamic transfer function models are shown to perform better than the benchmark static Greenshields models within the DTA framework, the scalability of the approach is not addressed as only four links are calibrated.

In recent years, the case has been made for the real-time calibration of microscopic traffic simulation models in an online setting, particularly with advancements in computational power and the emergence of richer datasets from sources like GPS equipped vehicles and connected vehicles. Microscopic traffic simulators, which model the behaviors of individual drivers in relation to the surrounding vehicles and network attributes, are frequently calibrated to reflect average field performance measures, though empirically-observed changes in individual driving behavior across time and driving environment have motivated new work in online calibration [32]. In particular, methods and justification for the online calibration of car-following models have been proposed. Hammit et al. [30] use vehicle trajectory data to calibrate four well-known car-following models, including the Gipps model and the Intelligent Driver Model, using a genetic algorithm approach. They use a sampling of data from the Strategic Highway Research Program's Naturalistic Driving Study as a proxy for trajectory data in their experimental results, and show the added benefit of using current traffic data for calibration instead of historical traffic data. While this provides empirical justification for the online calibration of microscopic traffic models, the results are primarily a proof-of-concept, and it is not clear how the calibration method would be implemented in a real-time setting. Papathanasopoulou et al. [50] also utilize

trajectory data for the online calibration of car-following models. They establish a rolling horizon framework for estimation and prediction of speeds and travel-times, and apply their approach to calibrating the Gipps car-following model. The algorithm is validated using real trajectory data from an experiment in Naples, Italy, involving four drivers in a platoon. Though they only calibrate the model for one vehicle in the case study, they also discuss methods that could be used to scale the technique, including decentralization and extrapolation of calibration results.

The majority of online calibration methods that focus on a specific set of parameters look at the dynamic origin-destination (O-D) matrices used as input for DTA systems. This problem grew out of the offline dynamic O-D estimation problem and has been adapted to the context of online calibration. It is frequently addressed in the literature because O-D matrices are a crucial input to demand models and show considerable fluctuation across time, even within a span of a few hours, making them an important component of online calibration. In addition, the number of O-D pairs in a road network can often number in the thousands, as mentioned previously, making this specific problem one of the most high-dimensional problems within the online calibration literature. As an example, for the Singapore network case study described in Chapter 5, there are over 4000 O-D pairs for which demand parameters must be calibrated for each 15-minute time step. At the same time, there is rich structural information relating O-D matrices to link flows (often the field data used for calibration) represented by the dynamic assignment matrix that can be exploited during online calibration through the use of analytical models. Toledo et al. [60] provide a detailed survey of research done on the broad topic of O-D estimation, from which this section summarizes the major contributions for real-time calibration.

The foundational approach for the O-D calibration problem relies on a generalized least-squares (GLS) formulation, proposed originally for offline estimation by Cascetta [21] and adapted for dynamic estimators of O-D matrices in 1993 [22]. The formulation estimates the O-D demand matrix from link counts using an analytical assignment model. Inaudi et al. [37] develop a sequential version of the model in order for it to be implemented in a real-time context. The extension also allows the

algorithm to be used for prediction, through a separate filtering approach combining historical and estimated O-D information. The generalized least-squares formulation has since been adapted for calibration and is a widely used benchmark for the O-D calibration problem. Bierlaire and Crittin [18] propose an efficient solution algorithm for the least-squares formulation by solving it online using the LSQR iterative algorithm, an analytical equivalent to the conjugate gradient method that exploits matrix sparsity [48]. In case studies using the DTA system DynaMIT on a small-scale network in Boston (10 O-D pairs) and a large-scale network in Irvine (627 O-D pairs), the LSQR algorithm significantly decreases computational effort compared to a Kalman filtering approach. In more recent results, Prakash et al. [51] apply a principal components analysis (PCA) approach for dimensionality reduction to the generalized least-squares formulation. The effectiveness of this modification is shown in a case study on the Singapore expressway network, and the results show a trade-off with a loss of 2% in estimation accuracy for a reduction in the problem size by a factor of 50 compared to the original full-dimensional generalized least-squares approach. Interestingly, the PCA version of the GLS formulation has better predictive accuracy than the full-dimensional method, which Prakash et al. attribute to the principal components better capturing correlations within the structure of the O-D matrices [51].

State-space modeling is also frequently applied to the O-D calibration problem. Ashok and Ben-Akiva [8, 9, 10] formulate the real-time O-D calibration problem as a state-space model and solve it using a Kalman filtering algorithm. Bierlaire and Crittin show theoretically that their approach is identical to the approach proposed by Ashok and Ben-Akiva when a Kalman filter is used to solve the least-squares formulation [18]. The approach by Ashok and Ben-Akiva makes two significant contributions to the literature. First, the state is defined as deviations of the O-D demand parameters from historical values, instead of the absolute values of the O-D parameters themselves. They argue this incorporates structural information about the spatial and temporal relationships between O-D flows contained in the historical estimates into the state-space model, though as a result this method also relies heavily on the

accuracy of the historical estimates as a baseline for online calibration [10]. Second, they propose augmentation of the state in the formulation to include O-D parameters from several prior intervals [9]. This allows for better modeling of longer-term temporal effects as O-D demand from previous time intervals does impact link count measurements, especially for a large-scale network. On the other hand, this increases the computational cost of the method by increasing the dimension of the state vector, which is another crucial component of use of the method for large-scale networks. These methods are validated on case studies with the Massachusetts Turnpike and an expressway network in Amsterdam, both medium-scale networks with limited route choice.

Since the earlier Kalman filtering work, there have been many variations aiming to incorporate more structural information into the problem, often utilizing additional sources of data from the network. Ashok [10, 11] proposes two other state-space model formulations, one based on a state comprising departure rates and destination node shares and the other based on a state that incorporates estimation of the dynamic assignment matrix as well. Lu and Zhou [38] propose an approach using an analytical representation of the cumulative flow diagram in the measurement equation to model congestion scenarios involving bottlenecks and incidents. Barceló [13] incorporates travel time data between sensors as measured using Bluetooth technology in place of a traditional assignment matrix in the measurement equation. The resulting model is solved using a Kalman filter, and empirical results are shown for two road networks in Spain using the AIMSUN simulator. Similarly, Lu et al. [39] incorporates turning volume data in an analytical model for dynamic assignment matrices, which is then embedded in a state-space model. In general, these proposed methods use specific types of data from the network being calibrated in order to better represent analytically the relationship between O-D demand and link counts. This in turn leads to more restrictive applicability of the models, and for case studies provided, scalability for large-scale problems is not shown as the networks used all have fewer than 300 O-D pairs, and often fewer than 100.

An approach proposed by Hashemi et al. [31] calibrates dynamic O-D demand

in addition to parameters for flow propagation models using a modularized framework. Through a series of consistency checks learned through a reinforcement learning framework, real-time adjustments are made to model parameters through a traffic flow propagation adjustment module and a demand adjustment module. Though multiple types of parameters are calibrated within the framework, the calibration is sequential and the modules operate independently. The performance of the calibration framework is shown with real-world data from a section of the US-75 highway corridor in Dallas, Texas, on a network of the highway and surrounding arterials and local roads.

Another interesting recent research direction has been determining a systematic way to isolate which O-D demand variables to calibrate in real-time. This approach is termed quasi-dynamic O-D demand estimation [23] and aims to impose more model structure on demand evolution within the day both to reduce the dimension of the problem for real-time implementation and to counteract poor prior demand estimation. Marzano et al. [41] posit that imposing constraints like the linear dynamic evolution of O-D demand and holding destination shares constant can lessen the issues caused by sparsity of measurement data in the O-D calibration problem. Extending this, Cascetta et al. [23] construct a quasi-dynamic O-D estimation framework using traffic count measurements, assuming constant O-D shares (the fraction of demand going to each destination node for a given origin node) over a longer period of time. They discuss statistical and empirical tests that can be used to confirm the validity of this modeling assumption, and apply a generalized least-squares algorithm under this framework to calibrate O-D demand for a motorway network in northeast Italy with 91 O-D pairs. Marzano [42] applies this framework to an extended Kalman filter approach for estimation and prediction of O-D demand. A promising application of this quasi-dynamic framework is in improving the prior O-D demand parameters, which can then be used as input for an online calibration approach.

The online calibration problem aims to tune simulator parameters, like those for stochastic traffic simulators as is the focus of this thesis, in order to better model and predict traffic behavior based on the most current information from the field. In that context, it falls within the broader research area of traffic state estimation

and prediction. Traffic state variables include link flows, densities, speeds, and travel times. Using available data, the traffic state estimation and prediction problem aims to model the state of traffic dynamics in a network at the current time and shortly into the future. While this thesis focuses on parameter calibration, approaches from the broader field of traffic state estimation and prediction that exploit analytical models for structural information have relevance to the aims of this thesis. The remainder of the literature review focuses on several of these approaches.

While much recent research in traffic state estimation and prediction has focused on developing machine learning methods like Bayesian networks [58, 71], neural networks [44, 63, 68], boosting [70, 74], and time-series models like ARIMA [64] (an introductory survey of these types of methods can be found in [62]), these methods require a large amount of historical data to train and generally do not exploit much problem-specific structure. On the other hand, there are several approaches that use analytical traffic flow models for traffic state estimation and prediction. A significant portion of these methods focus on traffic flow prediction for freeway networks based on analytical models like the Cell Transmission Model (CTM) [26] and other first-order flow models. State-space model formulations and Kalman filtering approaches are also very common. Nanthawichit [45] employs a macroscopic traffic flow model as the measurement equation of a state-space model where densities and space-mean speeds comprise the state. Link flows and speeds from probe vehicle data are used as observations. Anand [2] develops a similar approach, modified to accommodate the heterogeneous and less lane-disciplined traffic behavior of Indian freeway networks. An influential approach proposed by Wang and Papageorgiou [67] relies on a stochastic, second-order macroscopic flow model for freeway traffic. Again, a state-space model is formulated and solved using an extended Kalman filter. The estimation results in the case study presented show good performance for the freeway segment and boundary variables, and the time-dependent measurements fit well.

While the extended Kalman filtering approach is the most common method used in the literature to deal with the nonlinearity of the relationships in traffic networks, a disadvantage of the approach is its reliance on a linear approximation in the gradient

calculation. Alternative solution methods for state-space models, like particle filters [43], unscented Kalman filters [46], and ensemble Kalman filters [1], avoid linearization or any approximation of the gradient and have been applied to state estimation in freeway networks with good predictive accuracy for link densities, speeds, and flows. These methods deal with nonlinearities by use of sampling, though as a result, prove to be more computationally costly since more samples are needed for more intricate, higher-dimensional problems [46]. Further relaxations of the state-space model have also been proposed. Chen and Rakha [24] propose a particle filtering technique to predict travel times on freeway networks. Instead of specifying a state transition model, they employ trends in historical data to generate prior distributions at each time interval, which allows for the modeling of time-variant and non-Markovian dynamics. On the other hand, this approach requires a substantial amount of historical data and relies heavily on the ability of that historical data to model current traffic dynamics.

The state estimation and prediction approaches illustrate the limitations of an analytical approach. Many of the traffic flow models that provide the foundation for these approaches apply specifically to freeway networks and require particular types of data and information about boundary conditions, like entry and exit rates for on- and off-ramps to the network [49]. Some work has been done to extend the applicability of these model-based methods and to incorporate other data sources. Yang et al. [69] formulate a model for freeway traffic state estimation in Lagrangian space instead of Eulerian space to utilize newer sources of mobile data (i.e., GPS, Bluetooth, cell phone) in addition to more standard detector data. Other research has leveraged the emerging technology of connected vehicles as a data source. Bekiaris-Liberis et al. [14] employ a Kalman filtering approach for the estimation of density and flow on freeway networks, using only average speed measurements from connected vehicles and a minimal number of flow measurements from fixed sensors. Given the penetration rate of connected vehicles in the network, the density of mixed traffic on freeway segments can be estimated. Similarly, Emami et al. [28] apply a Kalman filter to traffic flow prediction using data from connected vehicles, but in the context of urban

arterial networks. In a very small case study across two intersections, the performance of the algorithm is measured across a variety of connected vehicle penetration rates, network signal settings, and demand arrival rates. As the technology becomes more widespread, they posit that connected vehicles will provide a low-cost means for obtaining data for traffic flow prediction applications.

In addition to the incorporation of novel data sources, model-based traffic state estimation and prediction algorithms have been extended to accommodate scenarios where previously established methods have had difficulty. Adverse weather conditions affect traffic dynamics on a network, primarily in terms of free-flow speed and link capacity, and should be modeled differently than normal conditions [17]. Bie et al. [17] adapt a second-order macroscopic traffic model for use in this context through the embedding of weather-specific fundamental diagrams parameterized by recent historical weather and traffic data. A case study using real data from a 10-kilometer freeway in Edmonton, Canada shows the added benefit of modeling weather effects on the estimation and prediction of speeds. Incident detection and its impact on the traffic state has also proven to be a difficult task in traffic state estimation. Wang et al. [65] propose an efficient particle filtering method that jointly estimates the traffic state and detects the location and severity of incidents. A multiple model particle filter is used to integrate the continuous macroscopic traffic model and the discrete incident model [66]. Similarly to the weather modeling extension, the impact of the incident on traffic dynamics is modeled through the fundamental diagram in the macroscopic traffic model in terms of the number of lanes open to vehicular traffic.

Another avenue for recent research in traffic state estimation and prediction has been algorithmic modifications to the standard Kalman filtering approach. The Kalman filter is built on strong modeling assumptions, such as the assumption of independent Gaussian noise in the transition and measurement equations of the state-space model. A couple of approaches aim to model processes outside of these assumptions. Guo et al. [29] use an adaptive Kalman filter, which continually updates the noise variances of the process, instead of using the standard Kalman filter assumption of constant variance across time. The approach produces point and confidence inter-

val forecasts for traffic flows on a freeway network, and empirical results show the largest improvement in cases where traffic is highly volatile. Cai et al. [19] develop a noise-immune Kalman filter approach for short-term traffic flow prediction. Whereas the standard Kalman filter provides the minimum mean square error estimates for a process with Gaussian noise assumptions, the noise-immune Kalman filter proposed is less sensitive to non-Gaussian noise in the process being modeled. One specific advantage of this modeling approach is its relative robustness to outliers in the measurement data. While the case studies shown are of low dimension, the development of a more flexible modeling framework is a promising contribution to the literature of traffic state estimation.

An interesting research direction in the literature is the development of hybrid algorithms that combine analytical models with models from machine learning. Allstrom et al. [1] apply an ensemble Kalman filter to a state-space model that fuses a CTM model with an autoregressive neural network to predict speeds. The hybrid approach improves travel time prediction for a small case study on a ring road network in Stockholm. For a problem setting on a larger network of arterial links in San Francisco, Hofleitner et al. [33] apply a similar hybrid approach incorporating traffic flow theory with a machine learning framework to estimate and predict travel time distributions. The aim of the approach is to take advantage of benefits of both techniques—robustness to noisy data due to the historical data used to fit the Dynamic Bayesian Network and creation of forecasts using traffic flow theory principles with physical relevance. In the case study, a small amount of streaming GPS probe data is able to produce strong performance in the prediction of travel times on a San Francisco network with 769 links.

1.3.1 Summary of existing literature

As seen in this literature review, there is a wealth of research on online calibration methods. Several approaches create frameworks for generic application to any type of traffic model, with the goal of increasing their applicability to many different problem contexts. These approaches also have the goal of jointly tuning parameters

across demand and supply models in order to describe the intricate nonlinear interactions with the simulator. While these methods provide a flexible framework for online calibration, they sacrifice computational efficiency and problem-specific structural information that is utilized by other algorithms in the literature. In both real-time origin-destination demand calibration and traffic state estimation and prediction problems, the incorporation of analytical models that represent relationships between parameters and measurements has been shown to both lead to accurate estimation and alleviate the computational burden of more numerical black-box calibration techniques [33]. In addition to these challenges, there is still a critical need to demonstrate the scalability of online calibration for large-scale networks. The vast majority of the case studies cited in the literature focus on problems of small or medium dimension (i.e., less than a few hundred parameters). For the online calibration of stochastic traffic simulators, an approach must be able to handle a problem with thousands of parameters with a reasonable computational budget to be of practical value for transportation practitioners.

1.4 Thesis contributions

This thesis makes several contributions to the state-of-the-art of online calibration. Most significantly, a new algorithm for online calibration is presented based on a state-space model framework solved by an extended Kalman filter. The algorithm provides a general-purpose and comprehensive framework in that the method is flexible and theoretically able to accommodate the online calibration of parameters of any type, and is able to incorporate multiple types of field data under a joint calibration set-up. The algorithm embeds information from an analytical (macroscopic) network model that is both tractable and differentiable. In doing so, it utilizes problem-specific network structure taken from the traffic simulator and the traffic network being calibrated to improve the performance of the algorithm, differing from black-box calibration algorithms in the literature. Additionally, the use of an analytical model reduces the computational cost of online calibration in comparison to other

extended Kalman filter approaches by eliminating the need for numerical gradient approximation. By decreasing the number of simulator runs required at each step of the calibration process, the algorithm provides an approach that will work with a limited computational budget. The performance of the online calibration algorithm proposed is demonstrated in several empirical case studies, through which its accuracy, scalability, and real-time performance are shown. It is directly compared to relevant benchmark algorithms, and the specific contexts in which it provides a competitive advantage are detailed.

More specifically, this thesis also develops a computationally tractable and differentiable analytical model for the special case of real-time O-D calibration that is able to model traffic conditions on a large-scale network. The analytical network model, which incorporates endogenous assignment, represents network attributes like O-D demand, densities, and travel times as a system of nonlinear equations. The system of equations scales linearly with the number of links in the network. The model presented is validated on two road networks, one small synthetic toy network and one large-scale real-world network, and its ability to replicate link observations like counts and densities is illustrated.

Finally, observations from the case studies are used to contribute to the ongoing discussion of algorithmic and practical considerations that must be taken into account for online calibration, both for the research community and practitioners. In the process, the advantages and disadvantages of different approaches are examined.

1.5 Thesis outline

The remainder of the thesis is organized as follows. Chapter 2 describes the proposed methodology for online calibration, starting from a general formulation of the online calibration problem. Chapter 3 presents the algorithmic details of the proposed approach along with a specific application to the problem of online O-D demand calibration. Chapter 4 presents a case study on the Florian toy network. Chapter 5 shows a more realistic application for the Singapore expressway network, in order to

analyze the performance of the algorithm on a real-world network. Finally, Chapter 6 concludes with the major findings and research considerations gleaned from the case studies. Possible future directions for exploration are also discussed.

Chapter 2

The Metamodel EKF Algorithm

The online calibration problem is presented in a general formulation in Section 2.1 with supporting information about the different components. The state-space model formulation is given in Section 2.2 along with definitions of the transition and measurement equations. In Section 2.3, the general Kalman filtering approach is first explained. Changes in the online calibration formulation and state-space model are given. Finally, the proposed algorithm is presented in relation to other Kalman filtering approaches from the literature.

2.1 Problem formulation

As noted in the previous chapter, stochastic traffic simulators are comprised of multiple supply and demand models, each with its own set of parameters to be calibrated. Although not all parameters need to be calibrated online, the interplay among models has a significant impact on the outputs of the traffic simulator. As a result, joint calibration of parameters is crucial and offers advantages over a sequential or iterative online calibration approach [3]. In addition, given the wide range of data sources available in a real-time setting, it is useful to create an online calibration framework that is adaptable to tuning parameters in response to multiple distinct types of field data. Offline-calibrated parameter values provide a useful foundation for good online calibration performance and can be incorporated into the online calibration problem

through use as a priori estimates. The influence of each of these separate components motivates a flexible and adaptable formulation of the online calibration problem.

In this thesis, the online parameter calibration problem is formulated as a nonlinear optimization problem:

$$\min_{\mathbf{x}^h \in \Omega} \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} (y_j^h - \hat{y}_j(\mathbf{x}^h))^2 + w \frac{1}{n} \left\| \mathbf{x}^h - \mathbf{x}_a^h \right\|^2 \qquad \text{for } h = 1, 2, 3, \ldots \tag{2.1}$$

The notation is defined as follows. A full list of notation used in the thesis is also included in Appendix A.

$\mathbf{x}^h$	vector of calibration parameters for time period h;
$\mathbf{x}_a^h$	vector of prior calibration parameter values for time period h;
y_j^h	observed field measurement j for time period h;
$\hat{y}_j(\mathbf{x}^h)$	expected simulator estimate of measurement j with input parameters $\mathbf{x}^h$;
w	weight parameter for prior information;
n	dimension of parameter vector $\mathbf{x}^h$;
Ω	set of constraints on the vector of calibration parameters;
$\mathcal{J}$	set of field measurements observed in real time.

For each time period h in the simulation period, the online calibration problem is represented as a nonlinear optimization problem with decision vector $\mathbf{x}^h$. The vector $\mathbf{x}^h$ can represent any combination of simulator parameters to be calibrated online, ranging from demand parameters like origin-destination (O-D) flows and route choice model parameters to supply parameters like link capacities and speed-density function parameters. Not all model parameters need to be included in the decision variable $\mathbf{x}^h$. The parameters to be calibrated in an online setting are often time-dependent; they are the subset of parameters that need to be tuned to adjust for time-dependent events like traffic incidents or weather that influence travel behavior in a shorter time frame than those considered during offline calibration. The vector of prior calibration parameter values $\mathbf{x}_a^h$ represents baseline estimates for the model parameters, typically provided by a previously-run offline calibration procedure.

The observed field measurements vector $\mathbf{y}^h$, with components y_j^h for each field measurement j in $\mathcal{J}$, can represent any set of surveillance data that captures prevail-

ing traffic conditions on the network, such as link flows, speeds, and densities. The set $\mathcal{J}$ is indexed by the number of available field measurements; different measurement types such as link counts and densities provided by the same sensor are treated and indexed separately. The measurements $\mathbf{y}^h$ are real-time surveillance data generated by ITS or similar means for each time period h. The most common source of real-time traffic data are often road sensors like inductive loop detectors, which provide point measurements like links counts and speeds.

The simulator estimates vector $\hat{\mathbf{y}}(\mathbf{x}^h)$, with components $\hat{y}_j(\mathbf{x}^h)$ for each measurement j in $\mathcal{J}$, are the corresponding traffic measurements output by the traffic simulator to be calibrated. As the simulator is stochastic, these estimates are often obtained via an average over multiple replications of evaluating the simulator at the same set of calibration parameters $\mathbf{x}^h$.

The above formulation aims to determine the optimal parameter values $\mathbf{x}^h$ at each time interval h that minimize the combined sum of squared differences between observed and simulated measurements in the traffic network and the distance from prior calibration parameter values. The first component of the sum given in Equation 2.1 represents the distance between real-time field measurements and those same network measurements as estimated by the simulator. Specifically, a set $\mathcal{J}$ of network measurements are gathered in real time from the field through sensors and other surveillance methods. The corresponding measurements from the simulator are estimated with calibration parameters $\mathbf{x}^h$ as input, and the online calibration problem aims to minimize the distance between the two vectors. The second component of the sum given in Equation 2.1 represents the distance between the decision vector and a set of prior calibration parameter values. As the online calibration problem is typically underdetermined, with the number of available field measurements an order of magnitude or more smaller than the number of parameters to be calibrated, prior information is used both to regularize the optimization problem and to center the search for parameter values around a set of previously-calibrated values. Both components of the sum in Equation 2.1 are normalized by the number of terms in the summation, given by the cardinality of the set $\mathcal{J}$ and the dimension of $\mathbf{x}^h$ respectively,

and the weight parameter w adjusts the relative importance of each component of the objective function.

2.1.1 Online calibration challenges

The online calibration of stochastic traffic simulators is a difficult problem due to the complexity of the supply and demand interactions modeled within the simulator. The relationship between the input model parameters and the output of the stochastic traffic simulator (e.g., link state estimates) does not have a closed-form expression. In addition, the relationship is often nonlinear, non-convex, and non-differentiable; thus, the objective function formulated in Equation 2.1 also has these properties.

For real-world traffic networks, online calibration becomes a high-dimensional optimization problem, as the number of parameters to be calibrated scales with network size. This is particularly true for problems involving calibration of time-dependent O-D matrices. Case studies presented in the literature solve online calibration problems for hundreds or thousands of parameters [18, 73]. The large-scale case study presented in Chapter 5 has a calibration parameter vector of dimension 4050. On the other hand, the set $\mathcal{J}$ of observed field measurements is usually low-dimensional—that is, the dimension of the measurement vector $\mathbf{y}^h$ used to adjust the parameters is typically an order of magnitude or more smaller than the calibration parameter vector $\mathbf{x}^h$ [73]. This leads to the online calibration problem being underdetermined. For the case study presented in Chapter 5, the dimension of the measurement vector $\mathbf{y}^h$ is 172.

Finally and most crucially, traffic simulators are computationally expensive to evaluate, which poses an issue for online calibration problems that must solve the optimization problem on a limited computational budget. To be functional for practitioners using these algorithms in a real-time setting, the online calibration problem must be solved every time interval within a rolling-horizon framework, often on the order of 5 to 30 minutes. These characteristics make online calibration a challenging optimization problem to solve, for which an accurate, but scalable and efficient, approach is needed.

2.2 State-space model

The online calibration problem is first formulated as a state-space model, an approach commonly found in the literature. The state-space model is a time-series model that describes the evolution mechanism over time and the observation mechanism for an underlying set of state variables. State-space models are defined entirely by the state vector, a system of transition equations that describe the evolution of the state through time, and a system of measurement equations that relate the state to observed measurements. For dynamic systems like the one solved in the online calibration formulation above, state-space model formulations are powerful restatements of the problem, as they provide a simple causal framework that explicitly represents movement of the system through time and ties underlying states to observable outputs. State-space models can be efficiently solved using recursive algorithms like Kalman filtering. In the following subsections, the various components of the state-space model formulation of the online calibration problem are provided.

2.2.1 State definition

The online calibration problem is formulated as a state-space model with the state vector defined to be the model parameters $\mathbf{x}^h$ to be calibrated at time interval h as defined in Equation 2.1. The dimension of the state vector is equal to the number of simulator parameters to be calibrated. Note that while many state-space model approaches in the literature define the state vector as deviations of the parameter values from historical parameter values [3, 18], the proposed state-space model defines the state vector as the absolute parameter values. The observed measurement vector is comprised of the field data $\mathbf{y}^h$ observed in real time at time interval h on the traffic network used during online calibration as defined in Equation 2.1.

2.2.2 Transition equation

The transition equation models the change in the state vector of calibration parameters from one time interval to the next. The transition equation for the proposed

state-space model is a simple random walk

$$\mathbf{x}^{h+1} = \mathbf{x}^h + \mathbf{u}^h \qquad \text{for } h = 0, 1, 2, 3, \ldots \tag{2.2}$$

where the error term has a Gaussian distribution $\mathbf{u}^h \sim \mathcal{N}(0, \mathbf{Q}^h)$ for all h with mean zero and covariance matrix $\mathbf{Q}^h$. As noted in Section 1.3, and to be discussed further in Section 6.2.1, the simple random walk transition equation may prove to be too reductive of a modeling assumption, in which case, the transition equation can be formulated as a higher-degree autoregressive process where the state at time interval $h + 1$ is a summation of linear transformations of the previous several states plus a random error term. In that case, the transition equation would have the form

$$\mathbf{x}^{h+1} = \sum_{q=h+1-p}^{h} \mathbf{F}_{h+1}^{q} \mathbf{x}^q + \mathbf{u}^h \qquad \text{for } h = p-1, p, p+1, p+2, \ldots \tag{2.3}$$

where $\mathbf{F}_{h+1}^{q}$ are matrices modeling the relationship between the state at time q and the state at time $h+1$ and the error term has a Gaussian distribution $\mathbf{u}^h \sim \mathcal{N}(0, \mathbf{Q}^h)$ for all h with mean zero and covariance matrix $\mathbf{Q}^h$. As formulated, each state $\mathbf{x}^{h+1}$ can be represented as a function of the previous states $\mathbf{x}^{h-p+1}, \mathbf{x}^{h-p+2}, \ldots, \mathbf{x}^h$.

2.2.3 Measurement equation

Measurement equations detail the relationship between information available for use in online calibration (field data, prior information) and the unknown parameter values that define the underlying state vector. The proposed state-space model uses two types of measurement equations, in an approach similar to the one formulated by Antoniou [3]. The first type of measurement equation gives the relationship between the underlying state and the observed field measurements like sensor counts, speeds, and densities; these are known as indirect measurements of the state. For the indirect measurement equations, the relationship between parameters and surveillance information is provided by the traffic simulator. The indirect measurement equation for

the proposed state-space model is

$$\mathbf{y}^h = \hat{\mathbf{y}}(\mathbf{x}^h) + \mathbf{v}_1^h \qquad \text{for } h = 0, 1, 2, 3, \ldots \tag{2.4}$$

where $\hat{\mathbf{y}}(\cdot)$ is the functional representation of the simulator and the random error term has a Gaussian distribution $\mathbf{v}_1^h \sim \mathcal{N}(0, \mathbf{R}_1^h)$ for all h with mean zero and covariance matrix $\mathbf{R}_1^h$. Note that the indirect measurement equation is nonlinear and has no closed-form analytical expression as it is given by the traffic simulator. The second type of measurement equation gives the relationship between the a priori values of the model parameters and the underlying state; these types of measurement equations are called direct measurement equations since they provide observations of the underlying state vector itself and preliminary estimates of the calibration parameters. The direct measurement equation for the proposed state-space model is

$$\mathbf{x}_a^h = \mathbf{x}^h + \mathbf{v}_2^h \qquad \text{for } h = 0, 1, 2, 3, \ldots \tag{2.5}$$

where the random error term has a Gaussian distribution $\mathbf{v}_2^h \sim \mathcal{N}(0, \mathbf{R}_2^h)$ for all h with mean zero and covariance matrix $\mathbf{R}_2^h$. Additionally, the random error terms $\mathbf{u}^h$, $\mathbf{v}_1^h$, and $\mathbf{v}_2^h$ are assumed to be independent.

2.2.4 Incorporation of prior information

As discussed previously, calibration often occurs in two stages starting with an offline component that provides a baseline for average demand and supply behavior throughout the simulation period followed by an online component. The goal of online calibration is to adjust this offline-calibrated prior information accordingly as real-time observations are gathered from the network. As such, the online calibration problem performs a more localized search of the parameter space. Prior information from offline calibration can be incorporated into the online calibration framework in several ways. Ashok and Ben-Akiva [9] proposed the concept of deviations for the O-D estimation and prediction problem, and Antoniou [3] expanded the use for a

general online calibration framework, where the state vector of the state-space model is formulated in terms of deviation from the offline calibrated or historical parameter values. The approach is justified as incorporating as much historical information into the formulation as possible by implicitly building temporal and spatial structural information into the state definition. In addition, the deviation state formulation is argued to better satisfy the Gaussian error assumptions of the Kalman filtering approach. On the other hand, there is little flexibility in the incorporation of prior information through deviations as regardless of the quality of the offline-calibrated parameter values, they are utilized in the same manner.

In the proposed state-space model, prior information is instead utilized in the form of additional direct measurement equations. The direct measurement equations provide a preliminary estimate of the calibration parameters; the offline-calibrated parameter values are precisely that. The use of direct measurement equations instead of deviations allows for more control of the influence the prior parameter information plays on the results of online calibration. The weight parameter w in the formulation given in Equation 2.1 can be adjusted according to the confidence placed in the prior information for the particular online calibration problem. While this leads to more algorithm parameters to tune to ensure correct algorithm outcomes, the added flexibility also allows for less reliance on the deviations structure.

2.2.5 Complete state-space model

In summary, the state-space model proposed for the online calibration problem is given as

$$\mathbf{x}^{h+1} = \mathbf{x}^h + \mathbf{u}^h \qquad \text{for } h = 0, 1, 2, 3, \ldots \tag{2.6}$$

$$\mathbf{y}^h = \hat{\mathbf{y}}(\mathbf{x}^h) + \mathbf{v}_1^h \qquad \text{for } h = 0, 1, 2, 3, \ldots \tag{2.7}$$

$$\mathbf{x}_a^h = \mathbf{x}^h + \mathbf{v}_2^h \qquad \text{for } h = 0, 1, 2, 3, \ldots \tag{2.8}$$

This set of transition and measurement equations completely specifies the state-space model, which is solved using Kalman filtering techniques, as described in Section 2.3.

The random error terms $\mathbf{u}^h$, $\mathbf{v}_1^h$, and $\mathbf{v}_2^h$ are assumed to have independent Gaussian distributions, and random errors are assumed to be uncorrelated across time intervals h. These strong assumptions allow for Kalman filtering techniques to be applied to solve the state-space model formulation of the online calibration problem, though variations of the standard Kalman filtering algorithm have been developed to relax the assumptions of the Gaussian random errors terms, for example in Chapter 4 of Chui and Chen[25] and as detailed in Section 1.3.

2.3 Proposed algorithm

The state-space model laid out for the online calibration problem provides a compact and efficient framework that describes the interactions of a dynamic system and allows for an efficient estimation of the underlying state. In particular, a Kalman filtering algorithm can be applied to the state-space model to find estimates for the state. In this section, the general Kalman filtering approach is explained, followed by a detailed description of the proposed extended Kalman filter (EKF) approach, coined the Metamodel EKF.

2.3.1 Kalman filtering

Kalman filtering is a powerful tool for traffic prediction that can incorporate the evolution of the traffic network through time considered in time-series models with physical relationships found in the traffic system. Kalman filters can be thought of as solving a hidden Markov model where the "hidden" state variables occupy a continuous space as opposed to a discrete state space [55]. They consist of a set of equations that describe the transition between hidden states of the Markov model and the distribution of observed outcomes based on the hidden state, both of which are built from the physical processes of the traffic network [54]. State estimation and prediction using Kalman filtering is a two-step process of "predicting" and "updating." During the prediction step, the Kalman filter produces estimates of the current state variables and their uncertainties from the estimates of the state variables at the

previous time step. In the update step, when a new data observation is recorded, it is used to update the previous prediction via a weighted average called the Kalman gain.

Kalman filtering yields the minimal mean square error state estimate for linear state-space models. At each time step h, the algorithm recursively updates two quantities—the underlying state estimate $\mathbf{x}^h$ and the associated covariance matrix $\mathbf{P}^h$ that specifies the uncertainty of the state estimate—according to the structure provided by the transition and measurement equations and through comparison of the current predicted measurements (based on the state estimate of the previous time step $\mathbf{x}^{h-1}$ and all previously observed measurements $\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^{h-1}$) with newly observed measurements $\mathbf{y}^h$. Because each step of the algorithm only relies on the previous state and covariance matrix estimates, estimation and prediction using Kalman filtering is computationally efficient in runtime and storage and can be done in real-time [54]. The recursive structure of the algorithm makes it a good approach for the online calibration problem.

For the state-space model formulation of the online calibration problem given in Section 2.2, the standard Kalman filter cannot be directly applied because the indirect measurement equation, $\mathbf{y}^h = \hat{\mathbf{y}}(\mathbf{x}^h) + \mathbf{v}_1^h$, is nonlinear. Generally, relationships in traffic networks are often nonlinear, and nonlinear variations of the Kalman filter are used instead, as has been described in Section 1.3. For the proposed Metamodel EKF algorithm, an extended Kalman filter (EKF) approach is used. The extended Kalman filter allows for nonlinear transition and measurement equations by using local linear approximations of the equations at each step in the algorithm. If the transition equation is nonlinear, a first-order Taylor series approximation of the equation centered at the current best state estimate is calculated and used in the time update step. Similarly, if the measurement equation is nonlinear, a first-order Taylor series approximation of the measurement equation around the current state estimate is calculated and used during the measurement update step.

The online calibration problem in its state-space formulation defined by Equation (2.8) has an auto-regressive (linear) transition equation, a nonlinear indirect

measurement equation, and a linear direct measurement equation. Thus, an extended Kalman filter can be applied to estimate the optimal calibration parameter values at each time interval h; no additional steps need to be taken with the transition equation, but a linearization step is added before the measurement update in the Kalman filter algorithm. The exact steps of the online calibration EKF algorithm are given in detail in Algorithm 1.

Algorithm 1 EKF for Online Calibration adapted from Antoniou [3]

Initialization:

$$\mathbf{x}^{0|0} = \mathbf{x}^0 \tag{2.9}$$

$$\mathbf{P}^{0|0} = \mathbf{P}^0 \tag{2.10}$$

for $h = 1$ **to** N **do**

Time Update:

$$\mathbf{x}^{h|h-1} = \mathbf{x}^{h-1|h-1} \tag{2.11}$$

$$\mathbf{P}^{h|h-1} = \mathbf{P}^{h-1|h-1} + \mathbf{Q}^h \tag{2.12}$$

Linearization:

$$\mathbf{H}^h = \frac{\partial \hat{\mathbf{y}}(\mathbf{x})}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\mathbf{x}^{h|h-1}} \tag{2.13}$$

Measurement Update:

$$\mathbf{G}^h = \mathbf{P}^{h|h-1}(\mathbf{H}^h)^\intercal \left(\mathbf{H}^h \mathbf{P}^{h|h-1}(\mathbf{H}^h)^\intercal + \mathbf{R}^h\right)^{-1} \tag{2.14}$$

$$\mathbf{x}^{h|h} = \mathbf{x}^{h|h-1} + \mathbf{G}^h \left[\mathbf{y}^h - \hat{\mathbf{y}}(\mathbf{x}^{h|h-1})\right] \tag{2.15}$$

$$\mathbf{P}^{h|h} = \mathbf{P}^{h|h-1} - \mathbf{G}^h \mathbf{H}^h \mathbf{P}^{h|h-1} \tag{2.16}$$

end for

The four main stages of the EKF algorithm are outlined below:

1. *Initialization*: Initial estimates $\mathbf{x}^0$ (Equation 2.9) and $\mathbf{P}^0$ (Equation 2.10) are made for the underlying state and the state covariance matrix, respectively.

2. *Time update*: At each time step h, the algorithm gives an estimate of the current state $\mathbf{x}^h$ and covariance matrix $\mathbf{P}^h$ based on the transition equation of the state-space model and the best state estimate at the previous time step $h-1$. As these predictions are made from the data available at time step $h-1$, they are

denoted by $\mathbf{x}^{h|h-1}$ (Equation 2.11) and $\mathbf{P}^{h|h-1}$ (Equation 2.12). This step is also known as the prediction step.

3. *Linearization*: The nonlinear measurement equation is approximated by its gradient at the current state estimate $\mathbf{x}^{h|h-1}$ (Equation 2.13).

4. *Measurement update*: At each time step h, newly observed measurements $\mathbf{y}^h$ are received. The state estimate $\mathbf{x}^{h|h-1}$ and covariance matrix estimate $\mathbf{P}^{h|h-1}$ calculated during the prediction step are updated to account for this new data. The expected measurement as given by the measurement equation based on the predicted state $\mathbf{x}^{h|h-1}$ is compared to the actual observed measurement $\mathbf{y}^h$, and an adjustment factor $\mathbf{G}^h$ called the Kalman gain is calculated (Equation 2.14). The predictions $\mathbf{x}^{h|h-1}$ and $\mathbf{P}^{h|h-1}$ are then corrected using the Kalman gain $\mathbf{G}^h$. The new state estimate that takes into account the observation $\mathbf{y}^h$ is made based on all data available at time step h and is denoted by $\mathbf{x}^{h|h}$ (Equation 2.15). Similarly, the new covariance matrix estimate that takes into account the observation $\mathbf{y}^h$ is denoted by $\mathbf{P}^{h|h}$ (Equation 2.16). This step is also called the correction step.

5. The prediction step and correction step are repeated at each time step h. The best estimate for the calibration parameters at time interval h is given by $\mathbf{x}^{h|h}$.

In addition to being a nonlinear function, note that $\hat{\mathbf{y}}(\cdot)$ does not have a closed-form analytical expression since it is given by the stochastic traffic simulator. Thus, analytical methods for deriving a first-order Taylor series approximation are not feasible. Instead, numerical differentiation methods have been proposed for the linearization step of the EKF as detailed in Section 1.3.

Two common numerical methods for estimating the gradient are central finite differences (FD) and simultaneous perturbation (SP). In both methods, simulator evaluations of slight perturbations near the current state estimate are used to approximate the gradient. For the central finite differences approach, a total of $2n$ additional simulator evaluations are needed for the first-order approximation, where

n is the dimension of the state vector $\mathbf{x}^h$. For each dimension of the state vector, the simulator is evaluated at two points near the current state estimate, a positive perturbation and a negative perturbation in that dimension. The simulated measurements at these two perturbations are then used to estimate the gradient in that dimension. Since the simulator is stochastic, these simulated measurements are impacted by the simulator error that can lead to noisy approximations of the gradient in that dimension. To reduce the impact of simulator error, multiple simulator replications at the same perturbed parameter values are often used and averaged; in this case, the number of additional simulator evaluations needed for the first-order approximation is multiplied by the number of replications used. For the simultaneous perturbation approach, all dimensions are randomly perturbed at the same time, following the conditions laid out by Spall [56]. The gradient approximation requires two additional simulator evaluations, regardless of the dimension of the state vector; one in the positive direction of the random perturbation vector and one in the negative direction. Simultaneous perturbation, while significantly faster and more scalable, sacrifices accuracy compared to central finite differences [5]. Due to the computational cost of numerically evaluating the gradient of the simulator, the linearization step of the extended Kalman filter algorithm is the most significant factor in the computational complexity of the algorithm, especially for high-dimensional online calibration problems. The use of a non-analytical measurement equation in the state-space model causes the algorithm to resort to numerical differentiation, which greatly impacts the speed and accuracy of the calibration algorithm.

In general, Kalman filtering provides a recursive and thus computationally efficient method for attaining the minimal mean squared error solution for linear state-space models. The extended Kalman filter allows for the method to be applied to the proposed state-space model with a nonlinear indirect measurement equation. The main drawback of the state-space model as currently formulated is the non-analytical measurement equation which requires numerical differentiation in the first-order Taylor series linearization step of the extended Kalman filter algorithm. The proposed Metamodel EKF algorithm takes a step around this issue; an analytical approximation of

the indirect measurement equation in the state-space model formulation is used as detailed in the next section.

2.3.2 Metamodel Extended Kalman Filter

In the proposed Metamodel EKF algorithm, an analytical approximation for the simulator is substituted for the actual stochastic traffic simulator in the state-space model. The analytical approximation, which is called the metamodel, provides a relationship between calibration parameters and observed measurements similar to the simulator; the difference is the metamodel is analytical, computationally tractable, and differentiable. This substitution in the state-space model, and in particular the indirect measurement equation given in Equation 2.4, leads to a completely analytical state-space model that can be solved using an extended Kalman filter algorithm without use of numerical differentiation methods for the linearization step. Eliminating the need for numerical gradient approximations involving multiple evaluations of the traffic simulator leads to an analytically tractable and less computationally costly algorithm.

In addition, the proposed metamodel contains problem-specific structural information from an analytical traffic model that relates the calibration parameters to the observed measurements. The problem-specific and model-driven information in the metamodel is embedded in the state-space model formulation with the aim of improving the quality of the parameter estimates found during the measurement update step of the extended Kalman filter algorithm. The general idea of formulating a metamodel with problem-specific information to build a computationally efficient algorithm has shown success in other continuous transportation problems like offline calibration [72] and congestion pricing [47].

In replacing the stochastic traffic simulator with a metamodel, the Metamodel EKF algorithm solves the following optimization problem, similar to the one presented

in Equation (2.1):

$$\min_{\mathbf{x}^h \in \Omega} \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} (y_j^h - m_j(\mathbf{x}^h))^2 + w \frac{1}{n} \left\| \mathbf{x}^h - \mathbf{x}_a^h \right\|^2 \qquad \text{for } h = 1, 2, 3, \ldots \tag{2.17}$$

with the same notation as defined for Equation (2.1) but where $\mathbf{m}(\mathbf{x}^h)$ is the metamodel estimate at time interval h of the observed measurements given calibration parameters $\mathbf{x}^h$ as input. The metamodel has components $m_j(\mathbf{x}^h)$ for each field measurement j in $\mathcal{J}$. If the metamodel $\mathbf{m}(\cdot)$ is a good approximation for the stochastic traffic simulator $\hat{\mathbf{y}}(\cdot)$, the parameter values computed as the solution to the optimization problem given in Equation (2.17) will be a good estimate of the optimal parameter values for the online calibration problem given by Equation (2.1). The solution to the optimization problem in Equation (2.17) at each time interval h is found using the Metamodel EKF algorithm found in Algorithm 2. The original state-space model formulation was given in Equation 2.8 with the state $\mathbf{x}^h$ defined to be the parameters to be calibrated. The corresponding state-space model solved by the Metamodel EKF algorithm has the same state vector and transition equation definitions, but the new indirect measurement equation is given by

$$\mathbf{y}^h = \mathbf{m}(\mathbf{x}^h) + \mathbf{v}_1^h \qquad \text{for } h = 0, 1, 2, 3, \ldots \tag{2.18}$$

with the same assumptions as in Equation (2.4) on the random error vector $\mathbf{v}_1^h$. This state-space formulation is completely analytical.

2.3.3 Algorithm details

The full Metamodel Extended Kalman Filter (Metamodel EKF) algorithm is given in Algorithm 2. The basic mechanism of the extended Kalman filter approach remains the same as in Algorithm 1, with time update and measurement update steps, but there are several key adjustments made to accommodate the metamodel approximation. First, some notation needs to be established. As mentioned in the previous section, the metamodel $\mathbf{m}(\cdot)$ is an analytical approximation of the stochastic traffic

simulator $\hat{\mathbf{y}}(\cdot)$. Each component $m_j(\cdot)$ of the metamodel, an approximation of field measurement j in the set $\mathcal{J}$, is a parametric function with metamodel parameters α_j^h, $\boldsymbol{\beta}_j^h$, and γ_j^h for time interval h. The metamodel also includes an analytical traffic model $\mathbf{q}(\cdot)$ with components $q_j(\cdot)$ that provides problem-specific structural information in the metamodel formulation. Further descriptions of these concepts are found in Section 3.1.

The Metamodel EKF algorithm has several different components. First, an offline phase is added to the EKF algorithm during which a database of parameter candidates are generated and evaluated with the simulator (and analytical model). These samples of the state vector $\mathbf{x}^h$ are accumulated in order to better approximate the traffic simulator with an analytical metamodel. In the algorithmic design, this phase takes place offline, which allows for samples to be collected according to a more general and free-ranging sampling procedure separate from the computational constraints of the online calibration problem. Discussion of what and how parameter candidates are chosen is found in Section 3.2.

Second, additional steps to the extended Kalman filter algorithm are needed to construct a local analytical approximation of the traffic simulator. To prioritize the local accuracy of the metamodel approximation near the current state estimate $\mathbf{x}^{h|h-1}$, the simulator is evaluated (and also the analytical model) a number of times at parameter candidates near or at the current state estimate $\mathbf{x}^{h|h-1}$ before the linearization step. The number of parameter candidates generated online depends on the computation budget of the online calibration problem. Once these points are generated, the metamodel parameters α_j^h, $\boldsymbol{\beta}_j^h$, and γ_j^h are re-fit with the set of parameter candidates collected so far to provide the most up-to-date approximation of the simulator. Details on the methodology of how the metamodel parameters are fit can be found in Section 3.2.

Once the metamodel has been updated, the quantities $\mathbf{q}(\mathbf{x}^{h|h-1})$ and $\left.\frac{\partial \mathbf{q}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{h|h-1}}$ are computed for use in the linearization and measurement update steps. The linearization step shown in Equation 2.23 calculates the gradient of the analytical metamodel $\mathbf{m}(\cdot)$ instead of the traffic simulator $\hat{\mathbf{y}}(\cdot)$. The calculation of the metamodel

gradient is detailed in Section 3.1. This gradient is then used to calculate the Kalman gain and update the state estimate to incorporate the new measurement $\mathbf{y}^h$ in the measurement update.

In summary, within the extended Kalman filter algorithm, replacing the traffic simulator with an analytical metamodel leads to the following changes:

- Many evaluations of the stochastic traffic simulator can be run offline. (It may be run online some number of times around the current state estimate to ensure better fit of the metamodel; this component of the experimental design can be adapted to the specific problem context.) The offline evaluations of the simulator are used to generate calibration parameter points with which to fit the metamodel during the online phase.

- At each time interval h in the EKF, the components $m_j(\cdot)$ of the metamodel are re-fit before the linearization and measurement update steps to the current set of simulator-evaluated points.

- The linearization step is done analytically for the fitted metamodel $\mathbf{m}(\cdot)$ instead of numerically for the traffic simulator $\hat{\mathbf{y}}(\cdot)$. Similarly, the measurement update is completed using the fitted metamodel instead of the simulator.

2.4 Conclusion

The Metamodel EKF algorithm proposed in this chapter aims to create significant gains in the efficiency and accuracy of online calibration methods by utilizing the network-specific structure of the problem. The algorithm solves the online calibration problem formulated in Equation (2.1) using a hybrid approach that embeds a metamodel approximation of the traffic simulator in place of the simulator within an extended Kalman filter approach. As detailed above, embedding this analytically differentiable metamodel approximation in the extended Kalman filter decreases the computational cost of the linearization step at each time step h. In the following chapter, details of the metamodel form and estimation are given. Examples of how the

Algorithm 2 Metamodel EKF for Online Calibration

Offline Phase

Offline Point Generation: generate a set of calibration parameter candidates $\mathbf{x}_1, \mathbf{x}_2, \ldots$, and evaluate with simulator $\hat{\mathbf{y}}(\cdot)$ and analytical traffic model $\mathbf{q}(\cdot)$ to create a bank of simulator-evaluated points $(\mathbf{x}_1, \hat{\mathbf{y}}(\mathbf{x}_1), \mathbf{q}(\mathbf{x}_1)), (\mathbf{x}_2, \hat{\mathbf{y}}(\mathbf{x}_2), \mathbf{q}(\mathbf{x}_2)), \ldots$

Online Phase

Initialization:

$$\mathbf{x}^{0|0} = \mathbf{x}^0 \tag{2.19}$$

$$\mathbf{P}^{0|0} = \mathbf{P}^0 \tag{2.20}$$

for $h = 1$ **to** N **do**

Time Update:

$$\mathbf{x}^{h|h-1} = \mathbf{x}^{h-1|h-1} \tag{2.21}$$

$$\mathbf{P}^{h|h-1} = \mathbf{P}^{h-1|h-1} + \mathbf{Q}^h \tag{2.22}$$

Online Point Generation: generate a set of local parameter candidates $\mathbf{x}_{h,1}, \mathbf{x}_{h,2}, \ldots$, close to $\mathbf{x}^{h|h-1}$ and evaluate with simulator $\hat{\mathbf{y}}(\cdot)$ and analytical traffic model $\mathbf{q}(\cdot)$

Metamodel Update: fit α_j^h, $\boldsymbol{\beta}_j^h$, and γ_j^h for $j \in \mathcal{J}$

Analytical Model Evaluation: compute $\mathbf{q}(\mathbf{x}^{h|h-1})$ and $\left.\frac{\partial \mathbf{q}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{h|h-1}}$

Linearization:

$$\mathbf{H}^h = \left.\frac{\partial \mathbf{m}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{h|h-1}} \tag{2.23}$$

Measurement Update:

$$\mathbf{G}^h = \mathbf{P}^{h|h-1}(\mathbf{H}^h)^\intercal \left(\mathbf{H}^h \mathbf{P}^{h|h-1}(\mathbf{H}^h)^\intercal + \mathbf{R}^h\right)^{-1} \tag{2.24}$$

$$\mathbf{x}^{h|h} = \mathbf{x}^{h|h-1} + \mathbf{G}^h \left[\mathbf{y}^h - \mathbf{m}(\mathbf{x}^{h|h-1})\right] \tag{2.25}$$

$$\mathbf{P}^{h|h} = \mathbf{P}^{h|h-1} - \mathbf{G}^h \mathbf{H}^h \mathbf{P}^{h|h-1} \tag{2.26}$$

end for

metamodel can be adapted for different calibration parameters and measurements are given, and the application of the proposed online calibration approach to the online O-D demand calibration problem is detailed at length.

Chapter 3

Metamodel Development and Algorithmic Considerations

In the previous chapter, a broad outline of the proposed Metamodel Extended Kalman Filter (Metamodel EKF) algorithm was described. In addition, the relationship between the Metamodel EKF approach and other extended Kalman filter approaches for the online calibration problem was delineated. In this chapter, the components of the Metamodel EKF algorithm are elaborated upon in full, particularly the formulation and fitting of the metamodel approximation, as well as practical considerations for the application of the proposed algorithm in various contexts. To illustrate how the general Metamodel EKF framework can be applied to a specific online calibration problem, the online origin-destination (O-D) demand calibration problem is formulated and an explicit implementation of the Metamodel EKF algorithm for the problem is given. A macroscopic analytical network model for the O-D calibration problem is presented and subsequently validated in the case studies presented in later chapters.

3.1 Metamodel formulation

The goal of the metamodel is to create a tractable and analytically differentiable approximation of the relationship between calibration parameters and network at-

tributes as represented by the stochastic traffic simulator $\hat{\mathbf{y}}(\cdot)$, and thus requires a formulation that can be adapted to any set of parameters or network measurements. Metamodels can generally be classified as either functional models, which are general-purpose functions that can approximate any arbitrary function (e.g., polynomials), or physical models, which are functions that incorporate problem-specific information. The metamodel formulation in the Metamodel EKF algorithm combines ideas from both functional and physical models, an approach that has seen success in offline calibration [72].

In order to achieve this, the formulation of the metamodel $\mathbf{m}(\cdot)$ makes use of information from a problem-specific macroscopic network model denoted by $\mathbf{q}(\cdot)$. The analytical network model is chosen to provide a computationally tractable and differentiable relationship between the calibration parameters and the observed field measurements and contains structural information about the traffic network being simulated. For each measurement j in set $\mathcal{J}$, the metamodel formulation is given by

$$m_j(\mathbf{x}^h, \alpha_j^h, \boldsymbol{\beta}_j^h, \gamma_j^h) = \alpha_j^h q_j(\mathbf{x}^h) + \phi(\mathbf{x}^h, \boldsymbol{\beta}_j^h, \gamma_j^h) \tag{3.1}$$

which gives the observed measurement j at time step h as a function of the calibration parameters $\mathbf{x}^h$ and metamodel parameters α_j^h, β_j^h, and γ_j^h. The $m_j(\cdot)$ are the components that make up the metamodel $\mathbf{m}(\cdot)$. Each $m_j(\cdot)$ gives an analytical approximation for the simulated observation $\hat{y}_j^h$ corresponding to field measurement y_j^h. Note that a separate metamodel component is formulated for each measurement j in set $\mathcal{J}$. The metamodel parameters α_j^h, β_j^h, and γ_j^h are re-fit at each calibration interval according to the method detailed in Section 3.2.

The metamodel formulation is comprised of two components. The first component $\alpha_j^h q_j(\mathbf{x}^h)$ emulates a physical metamodel by deriving problem-specific information from the analytical macroscopic traffic model denoted by $\mathbf{q}(\cdot)$ with components $q_j(\cdot)$. The expression $q_j(\mathbf{x}^h)$ represents the estimated measurement j found using the analytical traffic model with parameters set to $\mathbf{x}^h$. The traffic model provides the metamodel with a high-level analytical approximation of the relationship between

calibration parameters and observed measurements output by the simulator. The second component $\phi(\mathbf{x}^h, \boldsymbol{\beta}_j^h, \gamma_j^h)$ is a functional (i.e., general-purpose) term, with parameters $\boldsymbol{\beta}_j^h$ and γ_j^h, used as a local error correction around the calibration parameter values $\mathbf{x}^h$. This component emulates a functional metamodel and can be a polynomial or another general-purpose function of the calibration parameters $\mathbf{x}^h$. Depending on the problem setting, the functional component is adaptable to the needs and modeling constraints and can even incorporate problem-specific information (e.g., network topology, time-delay characteristics of the network) as well, so long as it is analytically differentiable.

The general form of the metamodel laid out in Equation 3.1 can be interpreted as a problem-specific analytical model of the simulator $q_j(\cdot)$ that is corrected by a scaling parameter α_j^h and an additive local correction term $\phi(\cdot)$. The aim of the analytical traffic model is to provide a good global approximation of the traffic measurement on the network, while the functional component of the metamodel is formulated to provide a good local approximation around the current state. Given the metamodel formulation in Equation 3.1, the gradient calculation found in Equation 2.23 of Algorithm 2 is

$$\frac{\partial m_j(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}^{h|h-1}} = \alpha_j^h \frac{\partial q_j(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}^{h|h-1}} + \frac{\partial \phi(\mathbf{x}; \boldsymbol{\beta}_j^h, \gamma_j^h)}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}^{h|h-1}} \tag{3.2}$$

where both $\frac{\partial q_j}{\partial \mathbf{x}}$ and $\frac{\partial \phi}{\partial \mathbf{x}}$ have analytical expressions since both are analytically differentiable.

3.1.1 Analytical traffic model

The analytical traffic model is a crucial component of the metamodel in that it is the primary contributor of problem-specific information, which is used to contribute structure to a general-purpose black box calibration method and improve the computational efficiency of the algorithm. While there is some flexibility in how the analytical traffic model is formulated, there are several requirements that must be satisfied. First, it must be an analytically tractable and differentiable model relat-

ing every calibration parameter in state vector $\mathbf{x}^h$ to every simulation-based network metric (with corresponding field measurement) in $\hat{\mathbf{y}}^h$. For calibration problems that involve intricate traffic behaviors on a large-scale urban road network, building a custom analytical model for the specific problem setting (e.g., choice of calibration parameters and performance metrics) can be difficult, and thus is a limitation of the proposed algorithm. The primary reason to use an analytical metamodel is to avoid the computational costs of numerical differentiation during the linearization step of the extended Kalman Filter algorithm; calculating the gradient of the analytical model (as well as solving the analytical model) should be computationally feasible. Second, in order for the Metamodel EKF approach to be scalable for high-dimensional problems and large-scale networks, the traffic model should also be scalable in terms of evaluation and differentiation.

As long as it satisfies these requirements, the analytical traffic model can incorporate a number of different concepts from traffic flow theory or queueing theory in its representation of the road network, as well as problem-specific information from the traffic simulator being calibrated. For the offline calibration problem considered by Zhang et al. [72], the road network is modeled as a probabilistic queueing network where each link is represented by a stochastic point queue. An example of an analytical traffic model for the online origin-destination (O-D) demand calibration problem is presented in Section 3.3.

3.2 Fitting the metamodel

Since the metamodel is used in the Metamodel EKF algorithm as a local analytical approximation of the simulator, it is essential that the procedure for fitting the metamodel to the traffic simulator provides the best local fit at the current state estimate given all available simulator information at each point in time. The fitting process is comprised of two steps—first information about the simulator-based relationship between calibration parameters and measurement outputs must be gathered, then the metamodel parameters must be fit to best approximate this relationship. As more

information from the traffic simulator is gained through the accumulation of simulator observations and as the current state estimate changes, the fit of the metamodel parameters should be updated.

3.2.1 Generating simulator-evaluated observations

Observations of the relationship between calibration parameters $\mathbf{x}^h$ and simulator outputs $\hat{\mathbf{y}}^h$ as governed by the stochastic traffic simulator are generated and accumulated to improve the fit of the metamodel approximation. These observations can be evaluated either in the offline phase or in the online phase immediately before estimating the metamodel parameters. The two phases of generating simulator-evaluated observations serve different purposes. In the offline phase, there is less of a constraint on the computational budget, and the simulator can be evaluated for a set of trial calibration parameters with the aim of improving the metamodel fit during the online phase. The sampling method for these trial points may be adapted to the problem-specific context, but should provide a bank of simulator-evaluated observations that span the space of realistic parameter settings. In the online phase, the computational budget is limited; the series of steps encompassing (1) generating simulator-evaluated observations, (2) re-fitting the metamodel parameters, (3) calculating the gradient of the metamodel must all occur in the time interval h, which is realistically on the order of 5 to 30 minutes. Any simulator-evaluated observations generated in the online phase are in the service of building a good local approximation of the simulator. As such, the parameter candidates should be focused on the sample space around the current state estimate.

Particularly for the offline phase, some consideration must be taken in how the trial points are evaluated by the traffic simulator, and how simulator outputs are measured. More specifically, the simulator-evaluated points may have been generated with different initial conditions than the current conditions of the simulator being approximated by the metamodel. There are a few options for generating these simulator-evaluated observations. In the first case, the traffic simulator may be evaluated in a time-independent sense. The initial conditions of the simulator are controlled

through the use of a pre-specified warm-up period and fixed warm-up demand. In using these points to approximate the simulator during online calibration, the resulting metamodel will be less reflective of the specific initial conditions at the current time interval but more points may be available with which to fit the metamodel. In the second case, the metamodel can be fit using only points generated during the online phase, with traffic conditions mirroring those in the traffic simulator. The exact starting conditions are replicated, but the metamodel is fit to fewer points and the computational benefit of evaluating simulator points offline is lessened. Finally, in the third case, all simulator-evaluated observations are used but a metric to characterize the similarity of initial conditions to the current conditions of the traffic simulator is used to weight the importance of each observation in fitting the metamodel. This concept of identifying and quantifying the similarity of different transient states of the traffic simulator may be an interesting direction for future analysis.

3.2.2 Estimating the metamodel parameters

Given a bank of simulator-evaluated observations, the parameters α_j^h, $\boldsymbol{\beta}_j^h$, and γ_j^h of each metamodel component $m_j(\mathbf{x}^h, \alpha_j^h, \boldsymbol{\beta}_j^h, \gamma_j^h)$ are estimated at time interval h to best fit the simulator before being used in the linearization and measurement update steps. At each time step h, the metamodel component for each measurement j is fit to the current set of simulator-evaluated points. By fitting the metamodel parameters at each time step, the approximation of the traffic simulator continues to improve with the availability of each new simulation evaluation. In addition, the metamodel parameters are updated to provide a local approximation of the simulator at the current state estimate $\mathbf{x}^{h|h-1}$. If at time step h the set of calibration parameters that have been evaluated by the simulator is given by $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ with corresponding simulator output for measurement j of $\{\hat{y}_j(\mathbf{x}_1), \hat{y}_j(\mathbf{x}_2), \ldots, \hat{y}_j(\mathbf{x}_N)\}$, the metamodel parameters α_j^h, $\boldsymbol{\beta}_j^h$, and γ_j^h are fit by solving the weighted least-squares minimization problem minimizing the distance between simulator estimates and metamodel

estimates:

$$\min_{\alpha_j^h, \boldsymbol{\beta}_j^h, \gamma_j^h} \sum_{i=1}^{N} \delta_i^h \left(\hat{y}_j(\mathbf{x}_i) - m_j(\mathbf{x}_i, \alpha_j^h, \boldsymbol{\beta}_j^h, \gamma_j^h) \right)^2 \tag{3.3}$$

where δ_i^h is the weight assigned to point $\mathbf{x}_i$ at time interval h. The weights δ_i^h are inversely proportional to the L^2 distance of the simulated point $\mathbf{x}_i$ to the current state estimate; that is, $\delta_i^h = 1/\left(1 + \left\|\mathbf{x}_i - \mathbf{x}^{h|h-1}\right\|\right)$. The weighted least-squares approach emphasizes calibration parameter points closer to the current state estimate $\mathbf{x}^{h|h-1}$ to prioritize that the metamodel is a local approximation of the simulator and its gradient. In the case where the number of simulator-evaluated points is fewer than the number of metamodel parameters being estimated, regularization is used to ensure the least-squares matrix is always full rank.

3.3 Online O-D demand calibration

In the previous sections, a general framework for the Metamodel EKF algorithm has been laid out including broad specifications of algorithmic steps and construction of the metamodel. In order to illustrate the flexibility of the approach to accommodate any set of calibration parameters and field measurements, the descriptions are given at a high level and the adaptability of the metamodel is emphasized. In this section, a specific implementation of the Metamodel EKF is provided for the online origin-destination (O-D) demand calibration problem. In applying the algorithm to this specific setting, concrete examples of the analytical network model, the metamodel formulation, and the metamodel fitting process are presented.

As summarized in Section 1.3, the online O-D demand calibration problem is a well-studied problem in the literature. In general, the aim is to identify time-dependent O-D matrices as inputs to the traffic simulator that minimize the distance between network attributes (often link counts) obtained from real-time field measurements and those output by the simulator. In the formulation given in Section 2.1,

the online O-D calibration problem is:

$$\min_{0 \leq \mathbf{x}^h \leq \mathbf{x}^{\max}} \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} (y_j^h - \hat{y}_j(\mathbf{x}^h))^2 + w \frac{1}{n} \left\| \mathbf{x}^h - \mathbf{x}_a^h \right\|^2 \qquad \text{for } h = 1, 2, 3, \ldots \tag{3.4}$$

The notation is defined as follows:

$\mathbf{x}^h$	vector of O-D demand for time period h;
$\mathbf{x}_a^h$	vector of prior O-D demand for time period h;
y_j^h	observed field measurement j for time period h;
$\hat{y}_j(\mathbf{x}^h)$	expected simulator estimate of measurement j with input O-D demand parameters $\mathbf{x}^h$;
w	weight parameter for prior O-D demand;
n	number of O-D pairs in $\mathbf{x}^h$;
$\mathbf{x}^{\max}$	upper bound vector for O-D demand;
$\mathcal{J}$	set of field measurements observed.

For calibrating O-D demand in an online setting, the most common field measurements used are link counts gathered from a set of sensors deployed through the road network. The formulation above allows for other types of field measurements as well. The prior O-D demand $\mathbf{x}_a^h$ in this scenario is often generated with an offline calibration algorithm and should provide a set of realistic (i.e., consistent with historical traffic patterns for the network) baseline parameter values. The weight parameter w can adjust the relative importance of the prior O-D demand values prior to the online calibration process, depending on the reliability or recency of the offline calibration values on a problem-specific basis.

3.3.1 Analytical traffic model for O-D calibration

In this section, an analytical traffic model for the online O-D demand calibration problem is formulated. This model provides an analytical and differentiable mapping of O-D demand parameters to network metrics, specifically link flows, densities, speeds, and travel times. For an online O-D demand calibration problem where the field measurements are a subset of these metrics, this analytical model can be used in the formulation of the metamodel. It also contributes to a general understanding

of the type of problem-specific structural information that can be incorporated in the metamodel approximation of the simulator.

The analytical model is implemented as a system of n nonlinear equations for a network with n links, which makes it scalable for high-dimensional online O-D calibration problems. The model considers the impact of endogenous traffic assignment (route choice probabilities are defined as a function of the O-D demands), which allows for a more realistic model, and is constructed using a multinomial logit route choice model and an approximation of the fundamental diagram relationship. These components introduce problem-specific information about network topology and traffic flow into the analytical model. While the model is solved as a system of n nonlinear equations, one for each link in the network, the following discussion breaks each nonlinear equation into six distinct components (represented by Equation 3.5 through Equation 3.10) for clarity's sake. Thus, for each link i in the network, the formulation of the analytical model is given by the following equations:

$$y_i = \frac{1}{n_i} \sum_{r \in \mathcal{R}_1(i)} P(r) d_{\mathcal{O}(r)} \tag{3.5}$$

$$P(r) = \frac{e^{\theta t_r}}{\sum_{j \in \mathcal{R}_2(\mathcal{O}(r))} e^{\theta t_j}} \tag{3.6}$$

$$t_r = \sum_{i \in \mathcal{L}(r)} \tau_i \tag{3.7}$$

$$\tau_i = \frac{\ell_i}{v_i} \tag{3.8}$$

$$v_i = v_i^{\max} \left(1 - \left(\frac{k_i}{k_i^{\text{jam}}} \right)^{\alpha_{1,i}} \right)^{\alpha_{2,i}} \tag{3.9}$$

$$k_i = c \frac{k_i^{\text{jam}}}{q^{\text{cap}}} y_i \tag{3.10}$$

Endogenous variables

d_s	expected hourly demand for O-D pair s;
y_i	expected hourly demand per lane for link i;
k_i	expected density per lane for link i;
v_i	expected (space-mean) speed for link i;
τ_i	expected travel time for link i;
t_r	expected travel time for route r;
$P(r)$	route choice probability for route r.

Exogenous variables

k_i^{jam}	jam density per lane of link i;
$v_i^{\max}$	maximum speed of link i;
q^{cap}	lane flow capacity;
ℓ_i	average lane length of link i;
n_i	number of lanes of link i;
$\alpha_{1,i}, \alpha_{2,i}$	fundamental diagram parameters of link i;
θ	travel time coefficient in the route choice model;
c	scaling parameter common to all links;
$\mathcal{O}(r)$	O-D pair of route r;
$\mathcal{R}_1(i)$	set of routes that include link i;
$\mathcal{R}_2(s)$	set of routes of O-D pair s;
$\mathcal{L}(r)$	set of links of route r.

The equations of the analytical model are discussed one by one. Equation 3.5 defines the expected hourly demand per lane on link i, y_i, as the sum of the expected route demand across all routes that include link i. The expected route demand is defined as the expected O-D demand for the O-D pair corresponding to the route multiplied by the probability of demand for the O-D pair choosing that route, given by the route choice probability $P(r)$. The route choice probability is calculated in Equation 3.6 using a simple multinomial logit model with utility functions that only depend on route travel time t_r. Note that the route choice set for each O-D pair is fixed; the set $\mathcal{R}_2(s)$ is exogenous. In Equation 3.7, the route travel time t_r is defined as the sum of the expected travel times for all links that comprise route r. The expected travel time τ_i of link i is calculated as the average link length, as averaged over all lanes of the link, divided by the expected space-mean speed of the link, denoted v_i (Equation 3.8). Equation 3.9 defines the relationship between expected speed and expected density per lane on link i. The equation is a differentiable approximation of

the fundamental diagram used by the mesoscopic simulator used in the case studies in the following chapters (DynaMIT-R). A differentiable approximation of the equation is used to maintain the differentiability of the analytical traffic model. In the final equation, Equation 3.10, the expected density per lane for link i, k_i, is assumed to follow a linear relationship with the expected demand per lane for link i, given as y_i. The ratio of the expected density k_i to the jam density k_i^{jam} is assumed to be equal to the ratio of expected demand y_i to flow capacity q^{cap}, to a scaling factor c common to all links. The scaling factor is fit based on validation experiments for the network. For the case studies detailed in the thesis, c is set to $\frac{1}{6}$.

Outputs of the analytical traffic model are obtained by simultaneously solving the system of equations for each link given by Equation 3.5 through Equation 3.10 for all links in the network. The analytical derivatives of each of the nonlinear equations have also been derived, so calculating the gradient is just a matter of evaluating another system of nonlinear equations. Given O-D demand parameters as inputs (given by $d_{\mathcal{O}(r)}$ in Equation 3.5) and specifying the exogenous variables, the analytical model solution derives estimates for network attributes that correspond to both simulator outputs and field measurements—link counts (y_i), densities (k_i), space-mean speeds (v_i), and route travel times (t_r). While the formulation represented by Equation 3.5 through Equation 3.10 separates the mapping of O-D demand to network attributes into six distinct equations, they can be consolidated into a single nonlinear equation per link. Thus, a road network with n links can be implemented as a system of n nonlinear equations. More specifically, it scales linearly with the number of links in the network and independently of network attributes like link length, the dimension of the route choice set, or the number of O-D pairs (the dimension of the online calibration problem). As a result, the analytical traffic model is suitable for online calibration on large-scale networks, as demonstrated in case studies in the following chapters.

3.4 Conclusion

In this chapter, the Metamodel EKF algorithm framework first detailed in Section 2.3 has been expanded upon to fill out important details on how the metamodel is constructed as an approximation of the traffic simulator. Necessary requirements for the metamodel formulation are given, as well as specifics on estimating metamodel parameters. To demonstrate how the Metamodel EKF algorithm can be used to solve an online calibration problem, the online O-D demand calibration problem is introduced. The implementation of the algorithm for the online O-D calibration problem is specified, of which the most significant contribution is the analytical traffic model for O-D demand calibration. In the next two chapters, the Metamodel EKF algorithm is used to solve the online O-D demand calibration problem for a synthetic toy network and a large-scale network for the city of Singapore.

Chapter 4

Florian Network Case Study

In the previous chapters, a novel algorithm for the online calibration of stochastic traffic simulators was proposed and developed in detail, and algorithmic specifications were discussed. In this chapter, the performance of the Metamodel EKF (MM-EKF) algorithm is evaluated in a case study on the Florian toy network, as measured both by the ability to identify parameter values that lead to simulator outputs similar to field measurements and by computational performance in a real-time context. The performance of the approach is compared to several other online calibration algorithms from the literature. Through this case study on a toy network, specific capabilities and advantages of the Metamodel EKF approach are highlighted. In particular, the Florian network case study aims to achieve the following three objectives.

First, the case study is used to validate the analytical model developed in Section 3.3. In doing so, the analytical traffic model relating O-D demand to link measurements is shown to reproduce traffic behaviors modeled by the simulator at a high level. The validation shows a close fit to sensor counts as measured by the correlation coefficient and as seen graphically; likewise, correlation with density measurements is strongly linear, though off by a constant factor for some sensors.

Second, the case study demonstrates the effectiveness of the algorithm for online calibration on a small-scale traffic network with performance comparable to several well-known benchmark algorithms in the literature. Performance is evaluated across multiple metrics, focusing specifically on the algorithm's ability to replicate observed

count and density measurements from the field. Metrics evaluating predictive performance are also analyzed and discussed, as these are a major argument for the importance of online calibration of traffic simulators.

Third, the case study evaluates specific contexts under which the Metamodel EKF provides a tangible advantage as compared to other online calibration methods. In particular, an experiment on the Florian network is conducted to demonstrate the flexibility of the Metamodel EKF approach to incorporate multiple measurement types (i.e., sensor counts and densities) in a unified online calibration framework. In this objective, the added capability of the proposed approach to perform calibration with multiple data sources is confirmed, whereas the generalized least-squares benchmark is only able to accommodate sensor counts as measurements. Additionally, the impact of simulator stochasticity on calibration performance is addressed in the comparison of the Metamodel EKF algorithm to other Kalman filtering methods utilizing different gradient estimation techniques. The trade-off between accuracy of the gradient estimation for black-box EKF methods and computational efficiency is made explicit, and empirical results show the benefits of the gradient estimation approach used by the MM-EKF algorithm.

The case study in this chapter focuses specifically on online O-D demand calibration for the Florian network. The online calibration algorithms have been implemented with the DynaMIT-R traffic simulator, Version 2.1.0 [15], and their performance across various demand scenarios is presented. Section 4.1 gives the specifications and experimental set-up of the case study, including a description of the network, the data generation process, performance metrics, and benchmark algorithms. Section 4.2 presents the experimental results in detail. Finally, Section 4.3 describes a specific experimental design used to evaluate the flexibility of the Metamodel EKF to incorporate and improve on the use of multiple data sources for online calibration.

4.1 Case study specifications

4.1.1 DynaMIT-R

For the case studies presented in this chapter and the following chapter, the online calibration algorithms are evaluated using Version 2.1.0 of DynaMIT-R, a Dynamic Traffic Assignment (DTA) system developed within the Intelligent Transportation Systems (ITS) Lab at MIT [15]. A high-level schematic of the DynaMIT-R system is provided in Figure 4-1 depicting the inputs, outputs, and various steps taken by the DTA system. DynaMIT-R relies on a real-time stochastic traffic simulator, which couples a detailed network representation with traveler behavior models in a rolling horizon framework to provide real-time traffic state estimation and prediction capabilities, as well as real-time generation of guidance information. As seen in Figure 4-1, the inputs to the simulator are the network representation and databases of historical information, along with a priori parameter values usually provided by an offline calibration phase, and surveillance information that provide metrics of network performance. A microscopic demand simulator component generates individual travelers and simulates their travel behavior in response to network information both pre-trip and en-route. A mesoscopic supply simulator component explicitly represents traffic dynamics like congestion, queueing, and spillback. The demand and supply simulator components are run in two main phases—state estimation and state prediction. The state estimation phase provides estimates of the current network state through metrics like O-D flows, link flows, queue lengths, and link speeds. Once the demand and supply simulators have been calibrated so that these network state estimates are congruent with the most recent information available from the surveillance system, the simulator creates predictions for the network state based on traffic behavior in the current state during the state prediction phase. The outputs of the state estimation and prediction phases are network performance metrics, which are then provided as input for the next time interval of the traffic simulator in a rolling horizon framework. The DynaMIT-R framework laid out in Figure 4-1 provides a good environment for an online calibration case study; the calibration module can be added to the state estima-

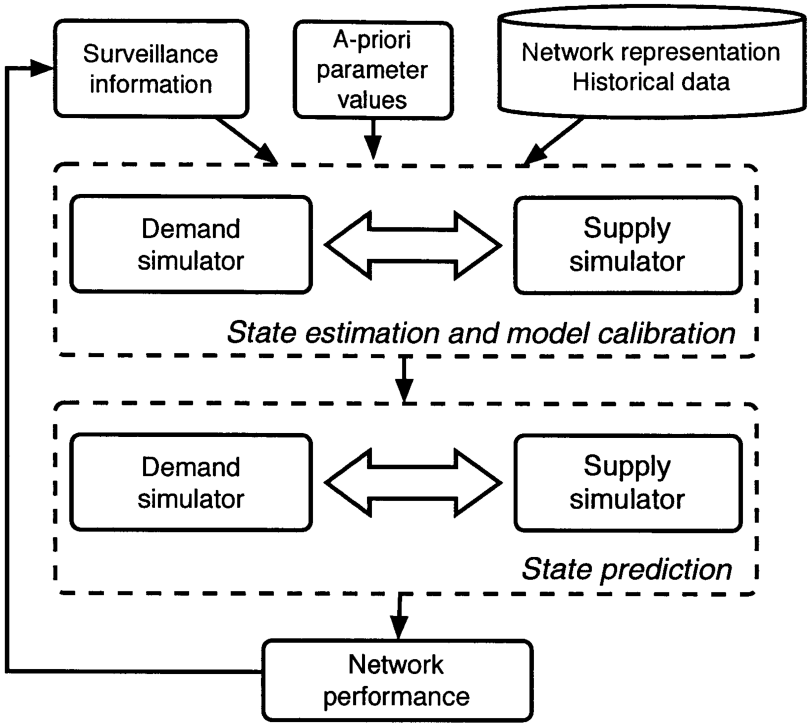

Figure 4-1: DynaMIT-R framework [3]

tion and model calibration phase. Among the various simulator inputs, DynaMIT-R provides the capability for specification of time-dependent origin-destination (O-D) demand matrices, which is the focus of the online calibration problem in this case study.

4.1.2 Florian network

The Florian network topology is shown in Figure 4-2. The Florian network is a simple synthetic toy network adapted from Astarita et al. [12]. The network includes 28 links with three origin nodes (labeled o_1, o_2, o_3 in Figure 4-2) and three destination nodes (labeled d_1, d_2, d_3 in Figure 4-2), for a total of nine origin-destination (O-D) pairs. Table 4.1 gives a summary of the O-D pairs and their corresponding origin and destination nodes. The network essentially consists of two parallel roads: one multi-lane highway with a higher flow capacity, free-flow speed, and a shorter free-flow travel time, and one arterial road with a single lane, a lower flow capacity, a lower free-flow

O-D Pair Index	Origin	Destination
1	o_1	d_1
2	o_1	d_2
3	o_1	d_3
4	o_2	d_1
5	o_2	d_2
6	o_2	d_3
7	o_3	d_1
8	o_3	d_2
9	o_3	d_3

Table 4.1: Summary of origin-destination pairs in the Florian network

speed, and a longer free-flow travel time. The highway route is accessible by a one-lane on-ramp and can be exited using a one-lane off-ramp. All links are one-way. For the origin-destination pairs 1, 2, 3, 4, and 7, there is only one possible route to choose from. For origin-destination pairs 5, 6, 8, and 9, vehicles can choose to take either the highway route or the local route. Free-flow route travel times for the network, calculated as the sum of the free-flow link travel times (link length divided by free-flow speed parameter) of links comprising the route, range from 3.9 minutes to 6.9 minutes. In comparison with the 15-minute time intervals set in the online calibration case study, the shorter free-flow route travel times indicate the impact of the time-dependent O-D demand will be observed in link measurements of the same interval with little time-delay. There are 12 link sensors spread throughout the network that are able to provide real-time traffic information (i.e., counts, densities). The sensors are identified with labels $S1, S2, \ldots, S12$, and their positions are indicated in Figure 4-2. For the experiments described, demand parameters are calibrated for all nine O-D pairs.

4.1.3 Data description

The objective of the Florian network case study is to calibrate O-D demand parameters for all nine O-D pairs across a typical morning peak period. In order to achieve this, a synthetic demand scenario is created for the simulation period of 3:00am to 10:00am. O-D demand is specified for every 15-minute period and mimics rush hour

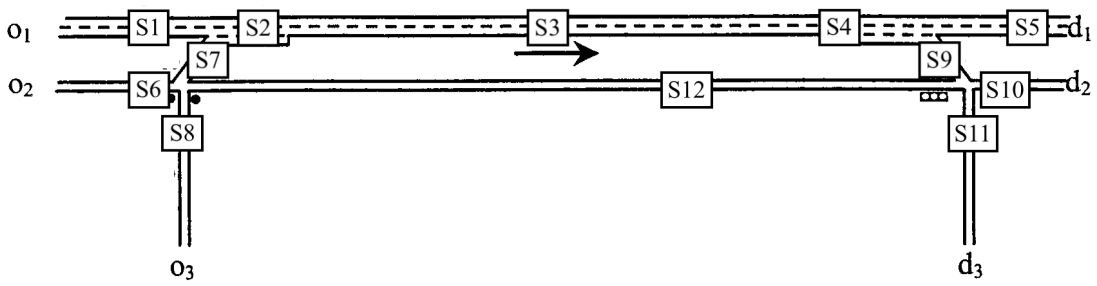

Figure 4-2: Florian network adapted from Astarita et al. [12]

traffic conditions, with total network demand increasing until the 15-minute period from 7:30am to 7:45am and then gradually decreasing from the morning peak. The O-D demand parameters represent the number of vehicles in each 15-minute time interval that depart from the origin node for the destination node for every O-D pair. Figure 4-3 shows the synthetic demand scenario for the Florian network. The x-axis spans the entire simulation period from 3:00am to 10:00am, and the y-axis plots the O-D demand for each 15-minute interval for each of the nine O-D pairs, labeled according to Table 4.1. In the demand scenario, there are three possible demand profiles each of the O-D pairs can follow—"High," "Medium," and "Zero" demand. O-D pairs 1 and 4 follow the "High" demand setting depicted by the blue line, which sees demand for each O-D pair start around 50 vehicles per 15-minute interval, increasing to a peak of approximately 250 vehicles per 15-minute interval, then decreasing to approximately 150 vehicles per 15 minutes. O-D pairs 2, 3, 5, and 8 follow the "Medium" demand setting depicted by the red line, which has a similar shape but at much smaller levels of demand for each O-D pair ranging from 25 vehicles per 15 minutes to 125 vehicles per 15 minutes. Finally, O-D pairs 6, 7, and 9 follow the "Zero" demand setting depicted by the yellow line and do not see any demand for the entire simulation period.

In addition to the synthetic "true" demand scenario, a set of prior O-D demands is constructed for the same simulation period of 3:00am to 10:00am to reproduce a realistic baseline of offline-calibrated demand parameters taken as input for the

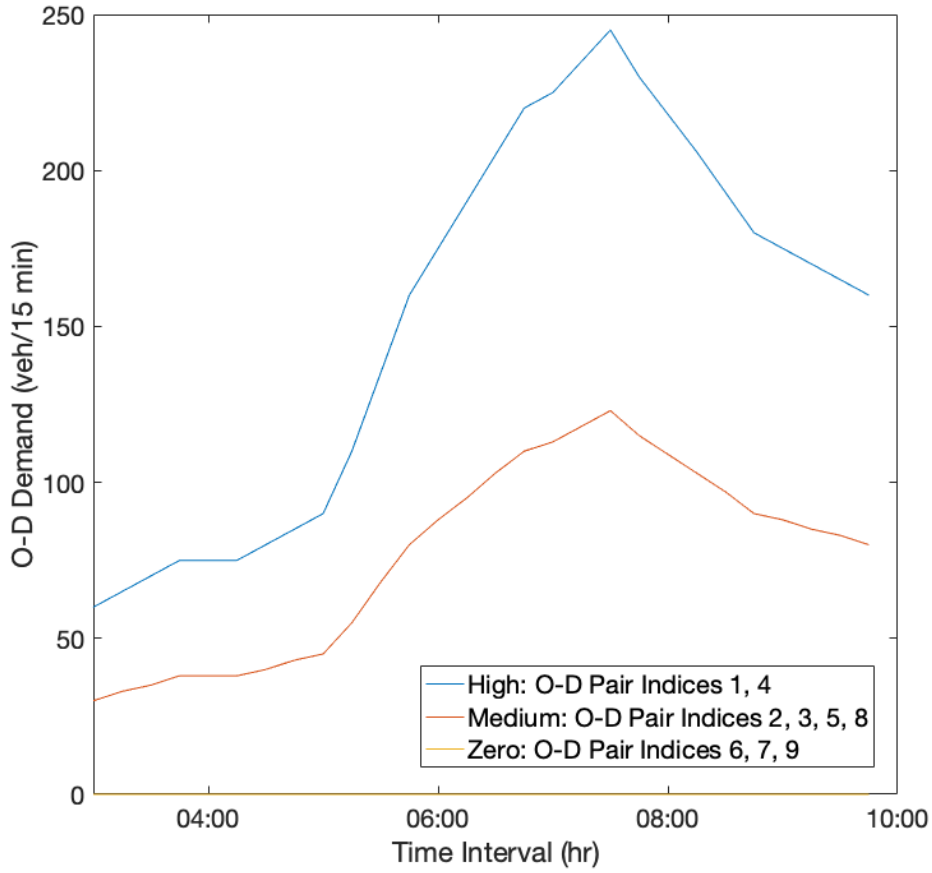

Figure 4-3: Synthetic O-D demand profile for Florian case study

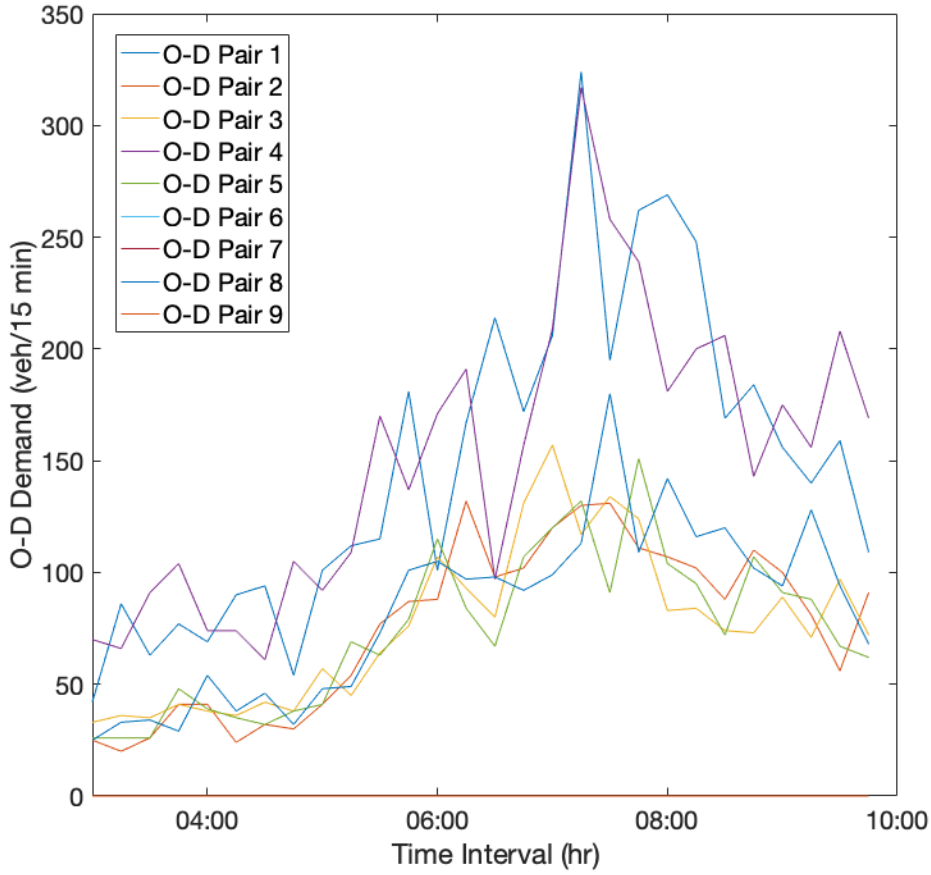

Figure 4-4: Synthetic prior O-D demand for Florian case study

online calibration problem. Each component of the synthetic prior O-D demand profile is derived from the "true" O-D demand scenario by adding an independent randomly drawn and normally distributed error term with an expectation of zero and a standard deviation of 20% of the "true" O-D demand value. Figure 4-4 shows the resulting synthetic prior demand for the Florian network. The axes and plotted lines have the same definitions as in Figure 4-3. As constructed, the trends of the prior demand are similar to those of the true demand with demand peaking at morning rush hour, as desired, but with substantial deviation due to the Gaussian noise to observe differences in sensor measurements between the two demand scenarios.

As both demand profiles are synthetic, sensor measurements observed in the field must also be computed synthetically. For each demand profile, sensor measurements

are created using the supply-only traffic simulator of DynaMIT (called DynaMIT-P [15]) by running 50 independent replications with the specified O-D demand parameters as input and calculating average simulator outputs across the 50 replications. Through this process, sensor count and density measurements for the 12 link sensors marked in Figure 4-2 are acquired and taken to be the "observed" field data for both the true demand and the prior demand scenarios.

Figures 4-5 and 4-6 plot the count and density sensor measurements for each 15-minute interval obtained from this process for a subset of the link sensors. Each subplot shows sensor measurements for one sensor, with the index number corresponding to the sensor labels in Figure 4-2. The x-axis spans the entire simulation period, and the y-axis measures sensor counts in vehicles per 15-minute interval in Figure 4-5 and sensor densities in vehicles per mile per lane in Figure 4-6. The blue solid line plots sensor measurements for the true demand profile, and the red dashed line plots sensor measurements for the prior demand profile. As seen in the figures, sensor measurements for the demand profiles show a similar trend to the underlying O-D demand. "Observed" counts for both true and prior demand scenarios increase until peak period then start to dissipate. Discrepancies between the true and prior demand scenarios are also greatest during morning peak period. Sensor densities show mild congestion on the network especially at Sensor 6 (particularly for the prior demand scenario) that dissipates after peak period.

4.1.4 Experimental design

In the implementation of the online calibration algorithms for the Florian network case study, several experimental specifications are made. All algorithms start with an initialization period of 15 minutes where O-D parameters are estimated using generalized least-squares (GLS). This estimate is used as the initial state $\mathbf{x}^0$ as defined in Equation 2.19 from Algorithm 2 in each of the Kalman filtering algorithms. Since the GLS only provides a single point estimate for the initial state vector, an ad-hoc covariance matrix is estimated for $\mathbf{P}^0$ to initialize the Kalman filter algorithms. The initial covariance matrix $\mathbf{P}^0$ is assumed to be a diagonal matrix, and each diagonal

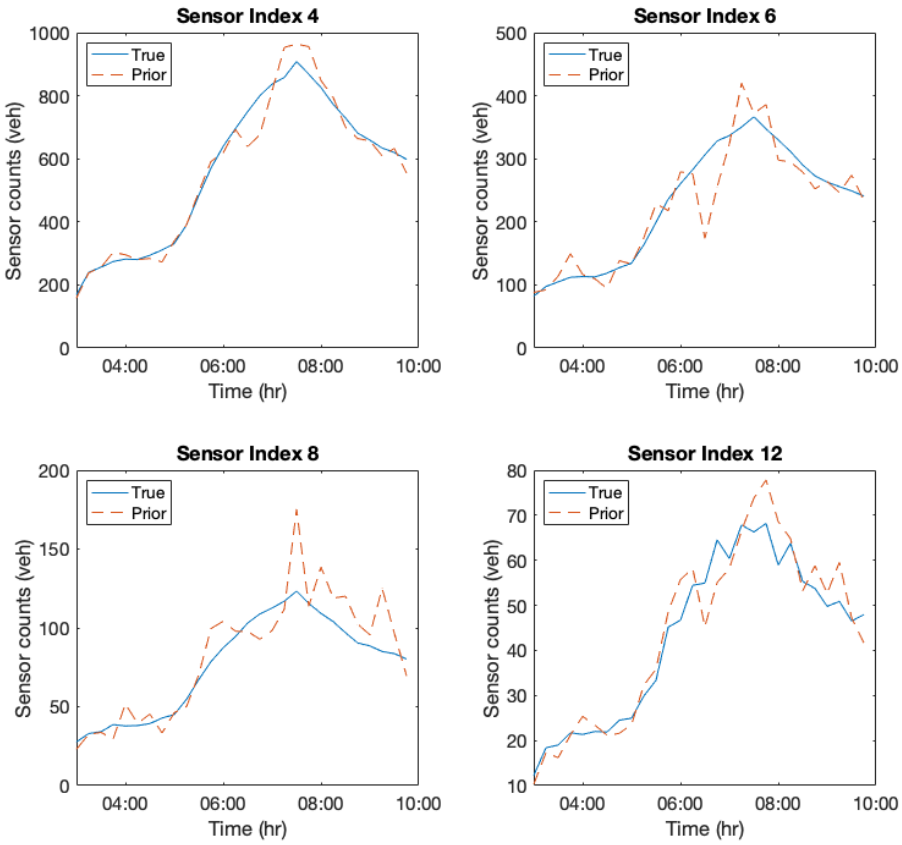

Figure 4-5: Synthetic sensor counts for Florian case study

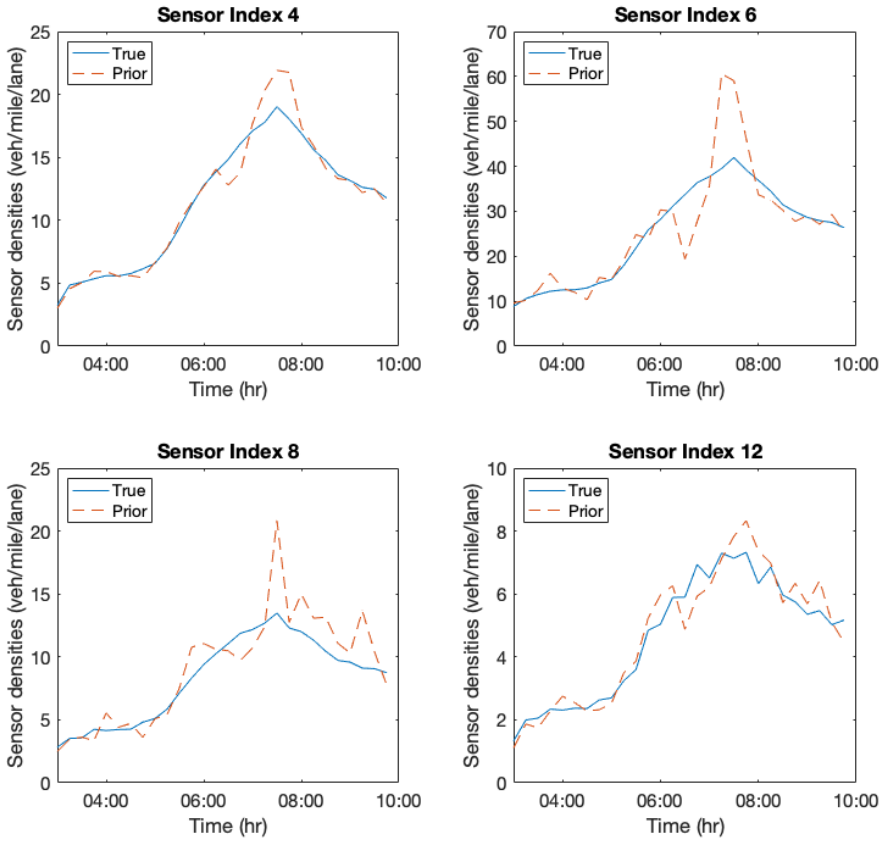

Figure 4-6: Synthetic sensor densities for Florian case study

entry (representing the variance of each O-D demand parameter estimate) is calculated as proportional to the magnitude of the corresponding initial state vector value. More specifically, $P_{i,i}^0 = \left(\max\left(a\left|x_i^0\right|, q_0\right)\right)^2$ where a and q_0 are tunable parameters of the Kalman filter algorithm.

For the underlying state-space model, a simple random walk with an auto-regressive degree of 1 is assumed for the transition equation. The indirect and direct measurement equations are specified in Section 2.2. Thus the state-space model for the case study is

$$\mathbf{x}^{h+1} = \mathbf{x}^h + \mathbf{u}^h \qquad \text{for } h = 0, 1, 2, 3, \ldots \tag{4.1}$$

$$\mathbf{y}^h = \hat{\mathbf{y}}(\mathbf{x}^h) + \mathbf{v}_1^h \qquad \text{for } h = 0, 1, 2, 3, \ldots \tag{4.2}$$

$$\mathbf{x}_a^h = \mathbf{x}^h + \mathbf{v}_2^h \qquad \text{for } h = 0, 1, 2, 3, \ldots \tag{4.3}$$

The random error covariance matrices for the transition equation ($\mathbf{Q}^h$) and measurement equations ($\mathbf{R}_1^h$ and $\mathbf{R}_2^h$) are fit offline from simulator data and shared by all online calibration algorithms using the state-space model formulation. Similar to the initial covariance matrix, the state-space model covariance matrices are parameterized by a and q_0 for $\mathbf{Q}^h$, b_1 and $r_{0,1}$ for $\mathbf{R}_1^h$, and b_2 and $r_{0,2}$ for $\mathbf{R}_2^h$. These six parameters of the Kalman filter algorithm, along with the weight parameter w from Equation 2.1, were tuned prior to the online calibration process using a simple grid search from a set of time-independent O-D demand scenarios. For each set of Kalman filter parameter values, the assignment matrix EKF algorithm detailed below in Section 4.1.5 was run to calibrate the time-independent O-D demand parameters; the parameter settings with the best fit to sensor counts for these offline experiments were $a = 0.4$, $q_0 = 1$, $b_1 = 0.03$, $r_{0,1} = 1$, $b_2 = 0.6$, $r_{0,2} = 1$, and $w = 100$. More discussion about the implications of the state-space model formulation can be found in Section 6.2.1.

For the proposed Metamodel EKF approach, the analytical network model detailed in Section 3.3 and a linear polynomial functional component are used in the metamodel formulation. An independent metamodel is fit for each simulator output corresponding to each of the available field measurements. For each field measurement

j in $\mathcal{J}$, the metamodel approximation is given by

$$m_j(\mathbf{x}^h; \alpha_j^h, \boldsymbol{\beta}_j^h, \gamma_j^h) = \alpha_j^h q_j(\mathbf{x}^h) + \left(\boldsymbol{\beta}_j^h\right)^T \mathbf{x}^h + \gamma_j^h, \tag{4.4}$$

for time interval h. For this case study, no offline points are simulated beforehand for the Metamodel EKF algorithm.

4.1.5 Performance benchmarks

The Metamodel EKF (MM-EKF) algorithm is compared to several online calibration benchmarks in the case study to separate out the relative performance of various aspects of the proposed approach. The first benchmark algorithm is the standard generalized least-squares (GLS) approach adapted from the dynamic O-D estimation problem [22]. As discussed in Section 1.3, the generalized least-squares algorithm is a well-known and effective method for solving the problem of online calibration of dynamic O-D matrices using link count measurements, though one restriction of the approach is its inability to incorporate other types of field measurements into its online calibration framework. It is often used as a benchmark in the literature [18, 51]. The GLS algorithm is chosen as a benchmark to compare the performance of the Metamodel EKF algorithm to an online calibration algorithm that does not rely on a state-space model framework.

The proposed algorithm is also compared to two extended Kalman filter algorithms for online calibration that use different methods for gradient estimation in the linearization step. The first, called the assignment matrix EKF (Assign-EKF), utilizes the empirical assignment matrix as an approximation of the gradient matrix [8]. The assignment matrix $\mathbf{A}^h$ provides a mapping of O-D demand flows to link flows in time interval h. For the online calibration of O-D demand parameters from link counts, the dynamic traffic assignment matrix provides a closed-form linear measurement equation in the state-space model: $\mathbf{y}^h = \mathbf{A}^h\mathbf{x}^h + \mathbf{v}_1^h$.

In the context of DTA systems, the assignment matrix is generated by the traffic simulator as an output of the dynamic network loading model [3, 51]. The entries

of the assignment matrix are empirically observed assignment fractions based on the route choice decisions made by simulated travelers. In the implementation of the Assign-EKF algorithm for this case study, the dynamic traffic assignment matrix $\mathbf{A}^h$ is treated as an exogenous variable provided by DynaMIT. As such, a standard Kalman filter can be applied since there is no need for linearization or computation of the gradient estimate. The main limitation of this approach is that it is only applicable for calibrating O-D demand parameters from link flow measurements, as this is the sole content of the dynamic assignment matrix. Additionally, the assignment matrix (i.e., gradient) is empirically derived, so if the simulator does not generate a potential route, the assignment matrix (gradient) will not represent it. As a result, this method relies heavily on all demand possibilities being simulated; divergence between the empirical and theoretical assignment fractions will introduce errors into the O-D calibration process, which will lead to biased and inconsistent estimates [11].

The second EKF algorithm used as a performance benchmark approximates the gradient using central finite differences of the simulator in the linearization step. This method was suggested in the original framework for the EKF approach proposed by Antoniou [3], but relies on a purely numerical and computationally costly method for gradient estimation. As discussed in Section 1.3, the number of simulator evaluations required to estimate the gradient at each time step scales linearly with the dimension of the underlying state, so for large-scale problems this method becomes unwieldy. In addition, the method of central finite differences is impacted by simulator noise when used to estimate the gradient of a stochastic traffic simulator. Other methods for numerical gradient estimation exist in the literature, specifically simultaneous perturbation [56], which shows a significant computational advantage compared to finite differences for large-scale problems [7]. At the same time, central finite differences is expected to provide a more accurate and reliable gradient estimation than simultaneous perturbation through the use of more simulator evaluations [7]. Thus for the toy network case study where the computational budget is less constricting, the finite differences method for gradient estimation is chosen over simultaneous perturbation, which also introduces noise from uncorrelated measurements and can be difficult to

tune properly [4]. The central finite differences benchmark method is referred to as the FD-EKF algorithm. The Assign-EKF and FD-EKF algorithms are chosen as benchmarks to compare the relative performance of the gradient estimation step in the Metamodel EKF algorithm to other gradient estimation methods. As the state-space model formulation and Kalman filter parameters are kept in common across algorithms, the only difference in these online calibration methods is in how they approximate the gradient in the linearization step.

Both the GLS and Assign-EKF benchmarks are specific approaches for the O-D demand calibration problem using sensor count observations. The FD-EKF benchmark is a general-purpose framework that can be used to calibrate a variety of different parameters using several different sources of field data, similar to the MM-EKF algorithm. For the initial case study, the only field data used is sensor count data, so all algorithms rely on the same inputs.

To get a sense of the effect of simulator stochasticity on each online calibration algorithm, each method is run three separate times, with the same random seed used to start each set of runs. While a set of three runs per algorithm gives some sense of the variation due to stochasticity, more runs are highly recommended in future analyses to reach conclusions with stronger statistical significance. In addition, all algorithms tested are compared to a baseline performance given by the prior O-D demand profile with no online calibration component, which represents the performance of offline-calibrated data in an online setting for this synthetic case study.

One metric for the performance of each algorithm that is assessed is fit to the observed field measurements, both in estimation and prediction. The goodness of fit is quantified using the root-mean-square normalized (RMSN) error metric, defined as

$$\text{RMSN} = \frac{\sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(\hat{y}_j - y_j\right)^2}}{\frac{1}{N}\sum_{j=1}^{N} y_j} \tag{4.5}$$

where N is the number of measurements, $\hat{y}_j$ denotes the estimated (or predicted) output from the traffic simulator for measurement j, and y_j denotes the true, or

observed, measurement j. Using the O-D calibration parameters estimated by each algorithm as inputs to DynaMIT-R, traffic state estimation and prediction data is collected as output from the simulator and compared to the corresponding data from DynaMIT-R output for the underlying synthetic true demand scenario. Since the only field data used for this case study is sensor count data, the RMSN error metrics for sensor counts, both for the current time interval and for 1-step and 2-step predictions, are used to evaluate the performance of the proposed MM-EKF algorithm against the benchmarks described above. As this demand profile is synthetic and the "true" sensor densities and O-D demand parameters are also known, the RMSN error metrics for both sensor densities and the underlying O-D demand parameters are also reported.

The root-mean-square normalized error metric is an often reported performance metric in the online calibration literature [6, 18, 52], as it is unitless and allows for comparison between datasets and models with different scales. Because the normalization factor is the average observed measurement, the error metric is also referred to as coefficient of variation of the root-mean-square deviation. In noting the performance of various online calibration algorithms in the literature, benchmark values for the RMSN error metric calculated for link counts reported as good performance in the literature range from 14% for GLS on an Irvine network case study [18], and for FD-EKF, 10-13% on a Southampton, UK case study [3] to numbers as large as 24-33% for the larger Singapore expressway network case study [52, 73], a variation of which is used as a case study in the following chapter.

4.1.6 Summary

To summarize, the online calibration experiments for the Florian network case study have the following attributes:

- Simulation period from 3:00am to 10:00am with 15-minute calibration intervals,
- State vector of calibration parameters is of dimension 9 consisting of O-D demand parameters,
- Measurement vector is of dimension 12 consisting of sensor counts,

- Algorithms run are the proposed MM-EKF and benchmarks GLS, Assign-EKF, and FD-EKF,
- Estimated and predicted (both 1 time interval into the future and 2 time intervals into the future) sensor counts are generated for all online calibration results,
- Performance is evaluated using the root mean-square normalized (RMSN) error metric.

4.2 Results

This section presents the results of the case study. Validation of the analytical traffic model on the Florian network is presented in Section 4.2.1 to illustrate the modeling power of the network-specific traffic model for O-D demand detailed in Section 3.3. Summary results for online calibration are then shown in Section 4.2.2 for the proposed MM-EKF approach and benchmark algorithms. A deeper dive into the algorithm's performance is then provided in Sections 4.2.3 and 4.2.5.

4.2.1 Validation of analytical model

The analytical traffic model for O-D demand detailed in Section 3.3 is validated on the Florian network for the time-dependent O-D demand scenario presented in Section 4.1.3. For each 15-minute time interval in the simulation period from 3:00am to 10:00am, the analytical traffic model is solved for link counts and densities with the input of O-D demand and then compared to the "true" sensor observations for the synthetic demand scenario output by the simulator. Figures 4-7 through 4-10 show the validation results graphically.

In Figure 4-7, analytical model counts are plotted on the y-axis and compared to "true" simulator counts (generated according to the process described in Section 4.1.3) plotted on the x-axis. Each point on the scatterplot represents a sensor count observation for a 15-minute interval in the simulation period. Each of the 12 link sensors

is plotted in a different color. The solid blue diagonal line, known as the "45-degree line", gauges what a perfect fit (correlation coefficient of 1) would look like. As can be seen on the scatterplot, the analytical traffic model is able to replicate the transient simulator counts very closely. The correlation coefficient for the analytical model and simulator sensor count datasets is 0.9992, extremely close to 1. This correlation can be seen in more detail in Figure 4-8, where the sensor counts for a subset of the sensors on the Florian network are plotted. Each subplot graphs the sensor counts over time for a particular link sensor. The x-axis spans the entire simulator period, while the y-axis measures vehicle counts. The "true" simulator counts are given by the solid blue line, while the analytical model counts are given by the dashed red line. For sensors 4, 6, and 8, the fit of the analytical model is near perfect; the analytical traffic model estimates the increase in flow to morning peak period followed by a slow decrease almost exactly. For sensor 12, the analytical model overestimates the flow on the link and does not seem to account for small fluctuations in flow during morning peak period. Even so, the analytical model follows the general shape of the simulator output.

Figures 4-9 and 4-10 give similar plots but for sensor density outputs. The setup of the two plots follows the same setup as for Figures 4-7 and 4-8. Whereas the fit for sensor counts was very close to a correlation of 1, Figure 4-9 illustrates that the relationship between simulated densities and analytical model densities more often differs by a constant factor depending on the sensor. The correlation for each individual sensor is still strongly linear, but correction using a scaling factor is necessary for some sensors. Figure 4-10 provides a similar conclusion. For each of the subplots, the general shape of the solid blue curve graphing simulator densities is replicated by the dashed red curve graphing the analytical model densities. Yet for sensor 4, the analytical model consistently overestimates simulator densities; the opposite is true for sensors 6 and 8. For sensor 12, the magnitude of the density measurements is correct, but the simulator density measurements show more noise over the simulation period than the analytical model. These scalar discrepancies indicate bias in the analytical traffic model, which can be corrected with further tuning of the exogenous parameters

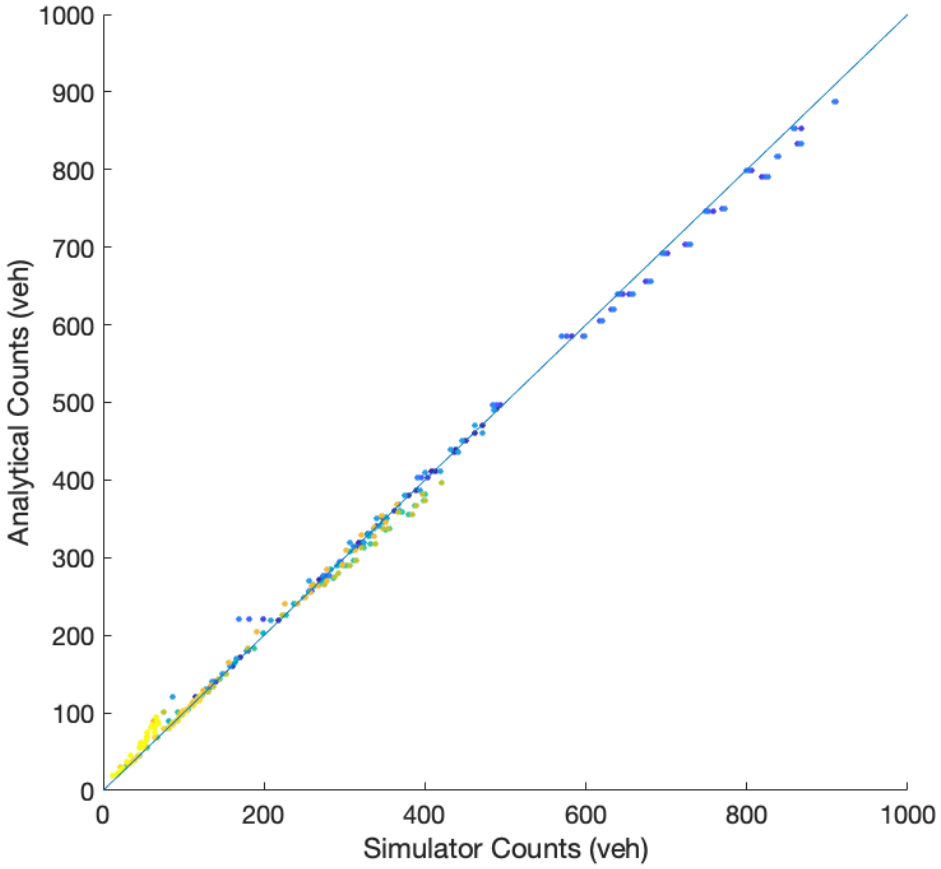

Figure 4-7: Scatterplot of simulated vs. analytical sensor counts for Florian network case study

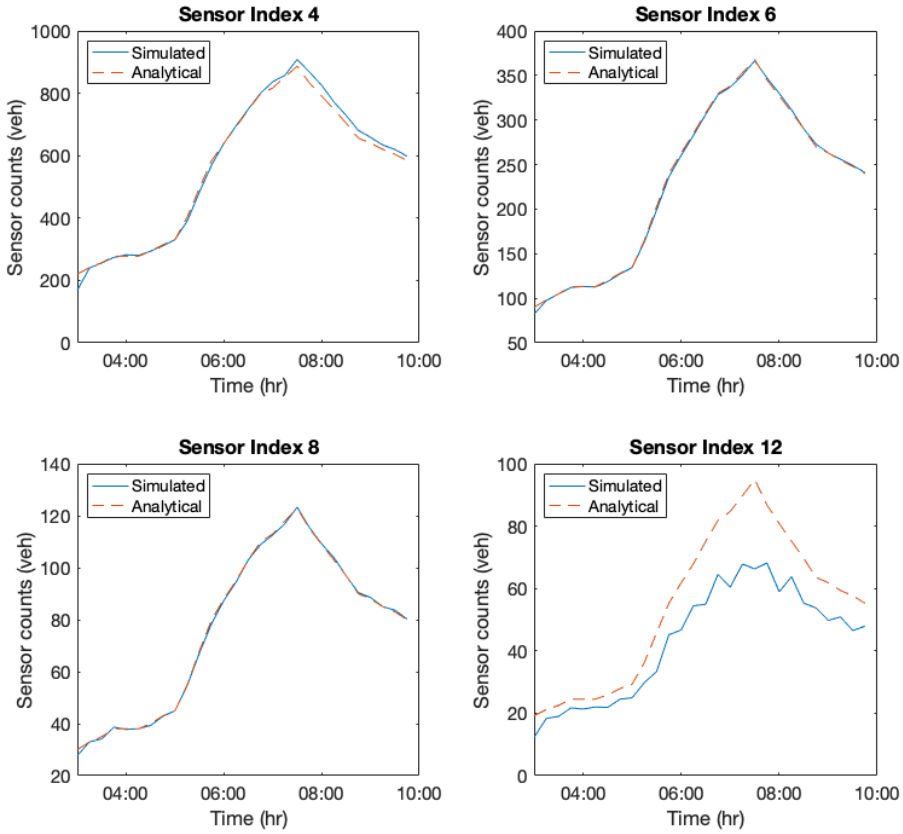

Figure 4-8: Comparison of simulated and analytical counts for subset of Florian network sensors

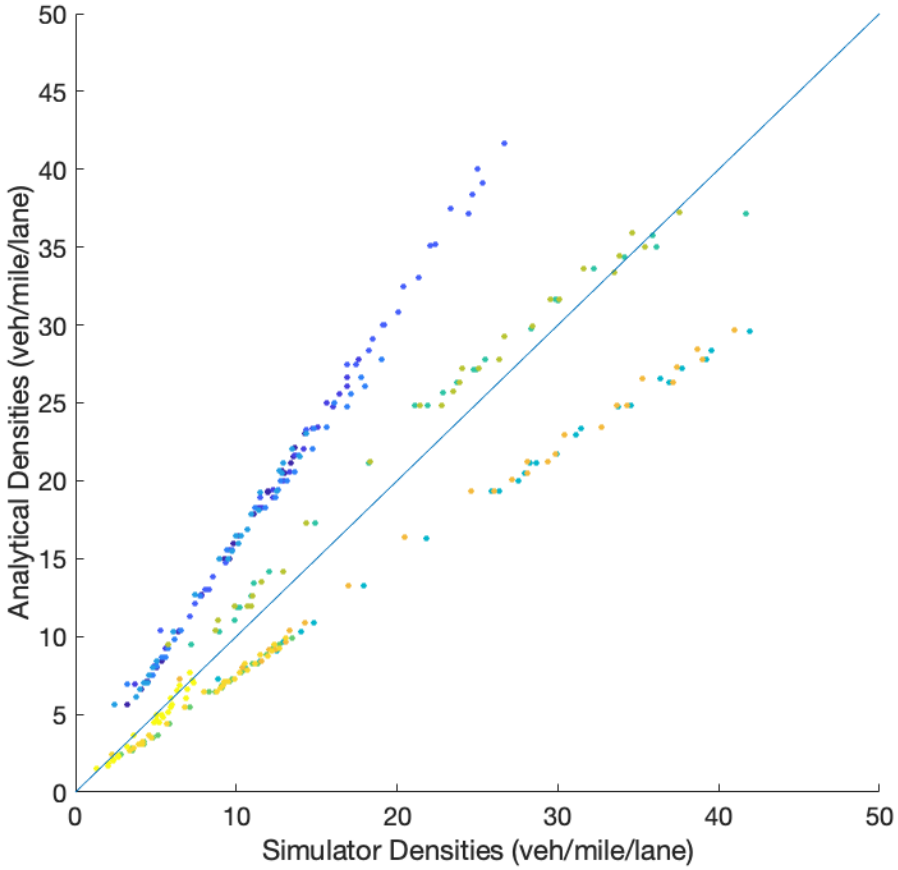

Figure 4-9: Scatterplot of simulated vs. analytical sensor densities for Florian network case study

of the model for the particular case study. In addition, this bias can be corrected in the metamodel formulation presented in Equation 3.1 using the parameter α_j^h. As a separate metamodel is fit for each measurement, the different linear relationships between the analytical model output and simulator output for sensor densities can be individually corrected during the fitting of the metamodel in the EKF algorithm.

4.2.2 Online calibration performance

Tables 4.2 and 4.3 show performance metrics for estimation of sensor counts, sensor densities, and O-D parameters across all online calibration experiment results. In Table 4.2, each row of the table reports results for one online calibration algo-

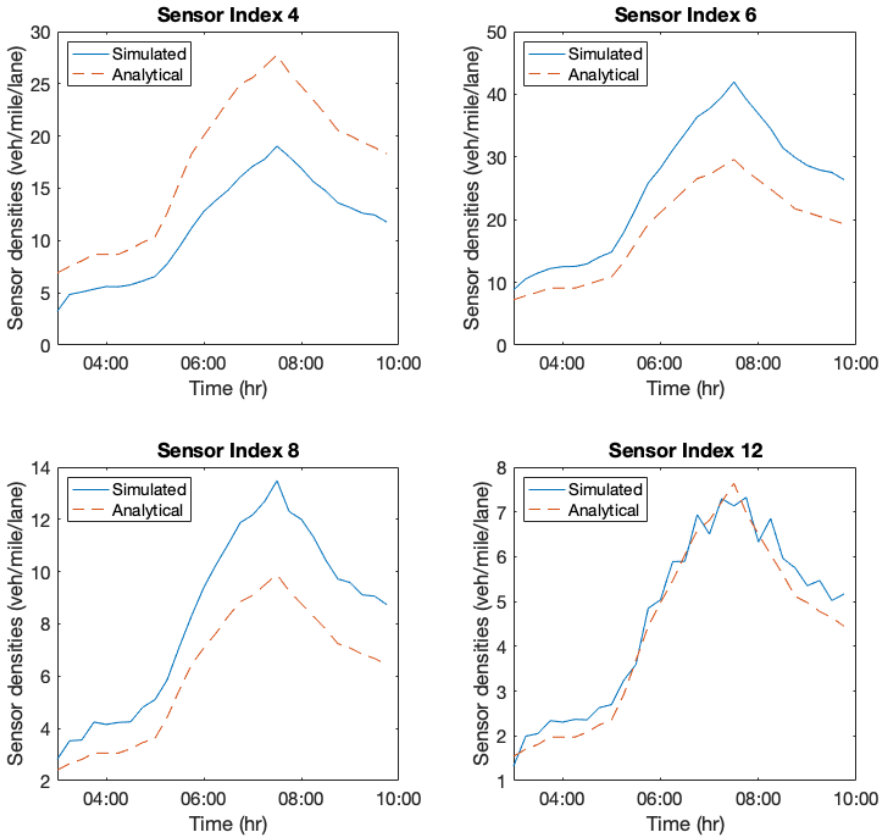

Figure 4-10: Comparison of simulated and analytical densities for subset of Florian network sensors

		counts		densities		O-D	
Algorithm	Run	RMSN	% improv	RMSN	% improv	RMSN	% improv
Baseline		0.091	-	0.326	-	0.243	-
MM-EKF	1	0.046	49.8	0.111	65.9	0.124	48.9
	2	0.048	48.1	0.106	67.4	0.137	43.5
	3	0.047	48.7	0.109	66.6	0.135	44.5
FD-EKF	1	0.055	39.6	0.123	62.2	0.161	33.8
	2	0.061	33.2	0.116	64.3	0.183	24.7
	3	0.062	32.4	0.120	63.1	0.179	26.3
Assign-EKF	1	0.049	46.1	0.122	62.7	0.146	39.9
	2	0.051	44.5	0.145	55.4	0.149	38.7
	3	0.047	48.1	0.126	61.5	0.137	43.7
GLS	1	0.028	69.6	0.138	57.7	0.154	36.7
	2	0.029	68.7	0.192	41.0	0.160	34.3
	3	0.031	66.6	0.178	45.4	0.163	32.8

Table 4.2: Estimation metrics for Florian network case study experiments

	counts		densities		O-D	
Algorithm	RMSN	% improv	RMSN	% improv	RMSN	% improv
Baseline	0.091	-	0.326	-	0.243	-
MM-EKF	0.047	48.9	0.109	66.6	0.132	45.6
FD-EKF	0.059	35.1	0.120	63.2	0.174	28.3
Assign-EKF	0.049	46.2	0.131	59.8	0.144	40.8
GLS	0.029	68.3	0.169	48.0	0.159	34.6

Table 4.3: Estimation results for Florian network case study averaged across three experiment runs

rithm run. "Baseline" reports error metrics for using prior O-D demand with no online calibration component as the estimated O-D parameters; the other algorithm runs are the proposed MM-EKF algorithm and the FD-EKF, Assign-EKF, and GLS benchmark algorithms, which are each run three times. The first set of two columns reports RMSN values for the fit to sensor counts. The next pair of columns reports the same metrics for fit to observed sensor densities, followed by a pair of columns for fit to the underlying true O-D demand parameters. The second column in each pair shows the percent improvement of the algorithm run results over the "Baseline" RMSN. Table 4.3 summarizes the results of Table 4.2. The table setup is the same, but the performance metrics reported for each online calibration algorithm are the average across three algorithm runs in Table 4.2. The percent improvement is again calculated off the "Baseline" RMSN.

The primary objective of the online calibration problem is to minimize the difference in sensor measurements output by the calibrated traffic simulator and sensor measurements observed in the field, as was formulated in Section 2.1. For the Florian network case study, the observed field measurements are sensor counts, so the primary performance metric is RMSN error for estimated sensor counts. As seen in Table 4.3, the estimation of sensor counts for the proposed MM-EKF algorithm gives a RMSN value of 4.7%, which is a 48.9% improvement over the baseline of prior O-D demand. A 95% confidence interval for the RMSN error metric for the MM-EKF algorithm as calculated using the results of the three algorithm runs as samples is $(4.5\%, 4.9\%)$, so the improvement over the baseline is statistically significant.

Results for the three benchmark algorithms also show improvement over the baseline in estimating sensor counts. Of the three, the generalized least-squares benchmark shows the greatest improvement over the baseline, with an average RMSN error of 2.9% which is an improvement of 68.3%. To determine if the difference in performance for the MM-EKF and GLS algorithms is statistically significant, a one-sided Mann-Whitney test is run using the three RMSN errors from the algorithm runs as the two samples for comparison. The Mann-Whitney test is a non-parametric test of the null hypothesis that the distributions of the two populations being sampled are equal. The Mann-Whitney test is used instead of a two-sample t-test in order to avoid making unnecessary assumptions about the population distributions of the error metrics. For all statistical tests of significance run in the case studies, the significance level used is $\alpha = 0.05$. For the comparison of estimation results for the MM-EKF and GLS algorithms, a one-sided test is run for the alternative hypothesis that the RMSN error metric for the GLS algorithm is less than the RMSN error metric for the MM-EKF algorithm, giving a p-value of 0.05 and indicating the result is significant.

The other two extended Kalman filter benchmarks also perform well compared to the baseline, with average RMSN errors of 5.9% for FD-EKF and 4.9% for Assign-EKF. As seen in Table 4.2, the variation in RMSN values for estimated sensor counts is small among runs of the same algorithm, which points to consistent performance and small replication variability. Again using one-sided Mann-Whitney tests for the alternative hypothesis that the MM-EKF algorithms performs better than the benchmark algorithms, the p-value for the comparison to the FD-EKF algorithm is 0.05 and the p-value for the comparison to the Assign-EKF algorithm is 0.1. Thus, in estimating sensor counts, the proposed MM-EKF algorithm shows better performance than FD-EKF but is statistically indistinguishable from the Assign-EKF benchmark. Overall, all four online calibration algorithms improve over the baseline with the GLS algorithm providing the best estimates for the observed sensor counts. For a small experiment on the Florian toy network, the estimation performance of the proposed MM-EKF algorithm has been validated and shown to be competitive with benchmark algorithms.

		1-step prediction				2-step prediction			
		counts		densities		counts		densities	
Algorithm	Run	RMSN	% improv	RMSN	% improv	RMSN	% improv	RMSN	% improv
Baseline		0.138	-	0.370	-	0.202	-	0.440	-
MM-EKF	1	0.120	13.1	0.189	48.9	0.146	27.5	0.250	43.2
	2	0.111	19.6	0.167	54.8	0.132	34.4	0.230	47.7
	3	0.116	16.1	0.196	47.1	0.141	30.4	0.240	45.4
FD-EKF	1	0.122	11.3	0.185	50.1	0.140	30.4	0.243	44.7
	2	0.120	13.3	0.170	54.0	0.146	27.8	0.239	45.7
	3	0.128	7.0	0.192	48.2	0.148	26.5	0.234	46.7
Assign-EKF	1	0.115	16.4	0.190	48.6	0.156	22.5	0.270	38.6
	2	0.115	16.3	0.185	50.1	0.145	28.3	0.245	44.4
	3	0.119	13.5	0.189	49.0	0.151	25.3	0.241	45.1
GLS	1	0.109	21.4	0.221	40.3	0.154	23.9	0.269	38.7
	2	0.119	13.9	0.267	27.9	0.156	22.9	0.283	35.6
	3	0.114	17.6	0.262	29.3	0.151	25.2	0.268	39.1

Table 4.4: Prediction metrics for Florian network case study experiments

	1-step prediction				2-step prediction			
	counts		densities		counts		densities	
Algorithm	RMSN	% improv	RMSN	% improv	RMSN	% improv	RMSN	% improv
Baseline	0.138	-	0.370	-	0.202	-	0.440	-
MM-EKF	0.116	16.3	0.184	50.3	0.140	30.7	0.240	45.4
FD-EKF	0.123	10.6	0.182	50.8	0.145	28.2	0.239	45.7
Assign-EKF	0.117	15.4	0.188	49.2	0.151	25.4	0.252	42.7
GLS	0.114	17.6	0.250	32.5	0.153	24.0	0.273	37.8

Table 4.5: Prediction results for Florian network case study averaged across three experiment runs

Tables 4.4 and 4.5 show experiment results for the prediction stages. The setup of both tables is the same as for Tables 4.2 and 4.3, respectively, except the columns now show performance metrics for one-step and two-step prediction for sensor counts and densities. Since the online calibration algorithms are run in 15-minute time steps for this case study, this means one-step prediction metrics correspond to fit of predicted measurements for 15 minutes into the future, while two-step prediction metrics correspond to fit of predicted measurements for 30 minutes into the future. In both tables, the raw RMSN values are reported as well as percent improvement for each algorithm run over the baseline case.

For the online calibration problem, RMSN error metrics for 1-step prediction and 2-step prediction are also crucial, as the ability to provide prediction and guidance capabilities in a real-time setting is a driving motivation for practitioners to imple-

ment online calibration methods. The Metamodel EKF algorithm shows an average RMSN of 11.6% for 1-step prediction and 14.0% for 2-step prediction, resulting in a percent improvement over the baseline of 16.3% and 30.7%, respectively. The benchmark algorithm that performed the best for estimation of sensor counts was GLS; here it shows an average RMSN of 11.4% for 1-step prediction and 15.3% for 2-step prediction. Noticeably, the performance metrics for prediction of sensor counts for MM-EKF and GLS are more similar than for estimation; the average 2-step prediction RMSN is actually smaller for the MM-EKF algorithm. Statistical tests bear out this conclusion—a Mann-Whitney test for 1-step prediction finds the two approaches statistically indistinguishable ($p = 0.35$). For 2-step prediction, the MM-EKF is shown to have better performance than the GLS benchmark, with a p-value of 0.05. The comparative advantage of the GLS benchmark in estimating sensor count observations is lost in the prediction stage. This may be due to overfitting by the GLS algorithm for the estimation phase, resulting in poor predictive performance, while the reliance of MM-EKF on state-space model assumptions leads to better prediction results.

The performance metrics for prediction of sensor counts for the other two EKF benchmarks show similar values to the MM-EKF algorithm. This similarity is confirmed by statistical tests—none of the comparisons of the Assign-EKF and FD-EKF algorithms to the MM-EKF algorithm produce statistically significant results. For 1-step prediction and 2-step prediction of sensor counts, the EKF algorithms all perform similarly, and all show a marked improvement over the baseline of prior O-D demand parameters. This indicates that regardless of the gradient estimation method, accuracy in the prediction stages is largely dictated by the EKF framework and the underlying state-space model.

Tables 4.2 through 4.5 also report results for two performance metrics that are not the direct objective of the online calibration problem—fit to sensor densities in both estimation and prediction and fit to the underlying O-D demand parameters. In the estimation phase, the proposed MM-EKF algorithm shows an average RMSN of 10.9% for fit to sensor densities and 13.2% for fit to the O-D demand parameters, for

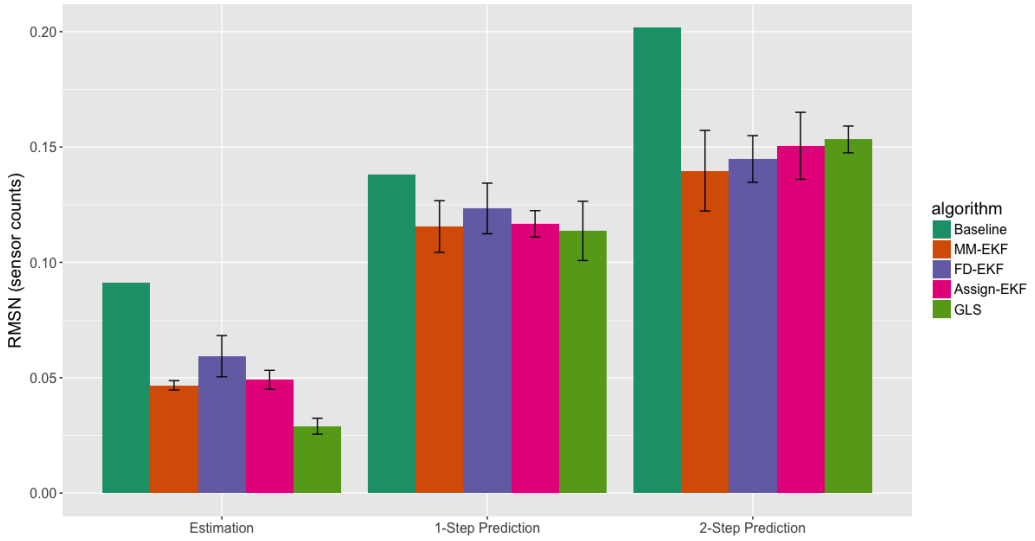

Figure 4-11: Average performance metrics for sensor count measurements for MM-EKF algorithm and benchmarks in Florian case study

improvements of 66.6% and 45.6%, respectively, over the baseline. Each of the three benchmark algorithms show a smaller percent improvement over the baseline with RMSN values ranging from 11.6% to 19.2% for estimated sensor densities across runs for all benchmark algorithms and RMSN values ranging from 13.7% to 18.3% for estimated O-D parameters. Statistical tests confirm the significance of these differences; the p-values for all but one of these pairwise comparisons to the proposed MM-EKF algorithm are 0.05. Only the fit to estimated O-D demand for the Assign-EKF and MM-EKF algorithms is statistically indistinguishable (p-value of 0.1). For prediction, the results are much more similar. All of the EKF algorithms show average RMSN values for 1-step prediction of sensor densities between 18% and 19%, while the GLS benchmark has an average RMSN value for 1-step prediction of sensor densities close to 25%. Overall, all online calibration algorithms show a significant improvement over the baseline case for both estimation and prediction of these secondary performance metrics (fit to sensor densities and O-D demand parameters). Whereas the GLS benchmark showed the greatest improvement in estimation of observed sensor counts, it performs less well for these secondary metrics, with the worst performance of the four online calibration algorithms in estimating and predicting sensor densities.

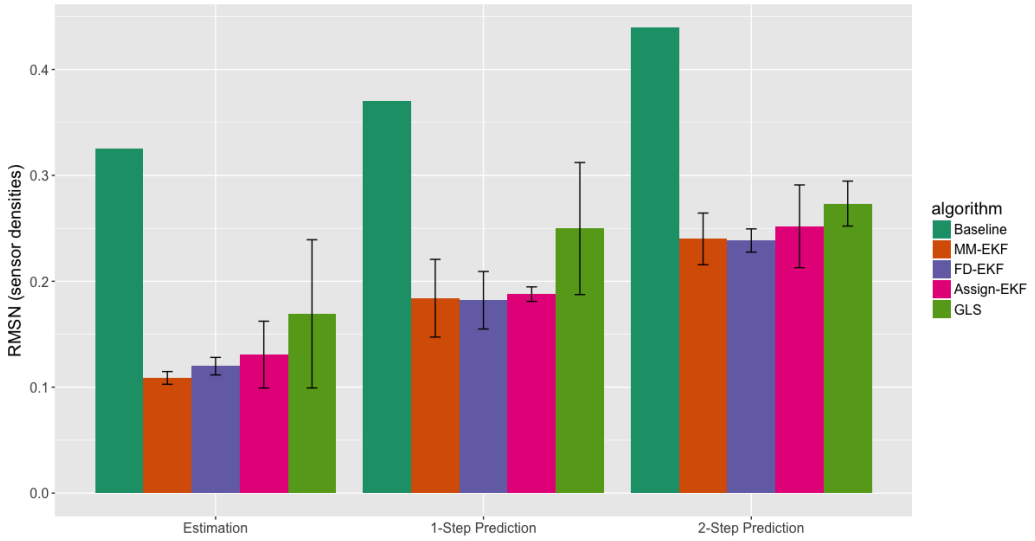

Figure 4-12: Average performance metrics for sensor density measurements for MM-EKF algorithm and benchmarks in Florian case study

Figures 4-11 and 4-12 show the experiment results graphically. Figure 4-11 summarizes the performance of the different online calibration algorithms for estimation, 1-step prediction, and 2-step prediction of sensor counts, as labeled on the x-axis. Each bar represents the average RMSN error across the three runs of that algorithm, given numerically in Tables 4.3 and 4.5. The error bars give the 95% confidence intervals as calculated from each three-run sample. Given the small sample size of three runs per algorithm, the confidence intervals reported here and in later results have a wide range, and more runs are recommended for a clearer statistical picture of the comparisons.

The conclusions reached above are supported by what is shown in Figure 4-11. All algorithms perform better than the baseline of no online calibration, with RMSN metrics for estimation well below 10% (a benchmark stated in Section 4.1.5 to signify good performance) and approaching the error level for stochasticity of the simulator. The MM-EKF algorithm also outperforms the FD-EKF benchmark even accounting for the 95% confidence intervals, although the GLS benchmark definitively performs the best overall. Even so, there is a relative deterioration in performance for the GLS algorithm going from estimation to prediction, signaling a level of overfitting

to sensor measurements that leads to worse predictive performance. All online calibration algorithms also improve on the baseline of prior O-D demand parameters by a statistically significant amount, as measured by the 95% confidence intervals, for 1-step and 2-step prediction.

Figure 4-12 gives the same type of bar graph as in Figure 4-11, but for performance metrics for sensor densities across estimation and prediction. As measured by the 95% confidence intervals in the figure, all online calibration algorithms statistically outperform the baseline of prior O-D demand parameters. For estimation of sensor densities, the 95% confidence intervals for all algorithms overlap; the same is true for 1-step and 2-step prediction metrics. The wider 95% confidence intervals of the GLS algorithm point to the higher variability among the three runs of the algorithm. As seen in Section 4.2.3, the worse performance of the GLS algorithm in estimating and predicting sensor densities is due to its inability to calibrate for increased congestion during peak period, while the other EKF algorithms perform comparatively to each other. As mentioned, the RMSN errors reported in Figure 4-12 show an auxiliary metric for performance that is not directly optimized for in the online calibration problem. Still, these auxiliary metrics can provide a sense of whether the online calibration algorithms are just overfitting to the field observations or ultimately estimating the true parameters of the underlying traffic state.

4.2.3 Results by time interval

To further analyze the relative performance of the various online calibration algorithms in the Florian network case study, several performance metric results are decomposed into performance for each hour of the simulation period. The hourly breakdown provides a better map of the temporal aspects of each algorithm in calibrating the Florian demand scenario from starting at low congestion through the morning peak period. Figures 4-13 through 4-16 gives average RMSN error metrics for estimation and prediction of sensor counts, and estimation of sensor densities. The RMSN values were calculated by averaging RMSN values across the four 15-minute time intervals comprising each hour, and then averaging across all three runs

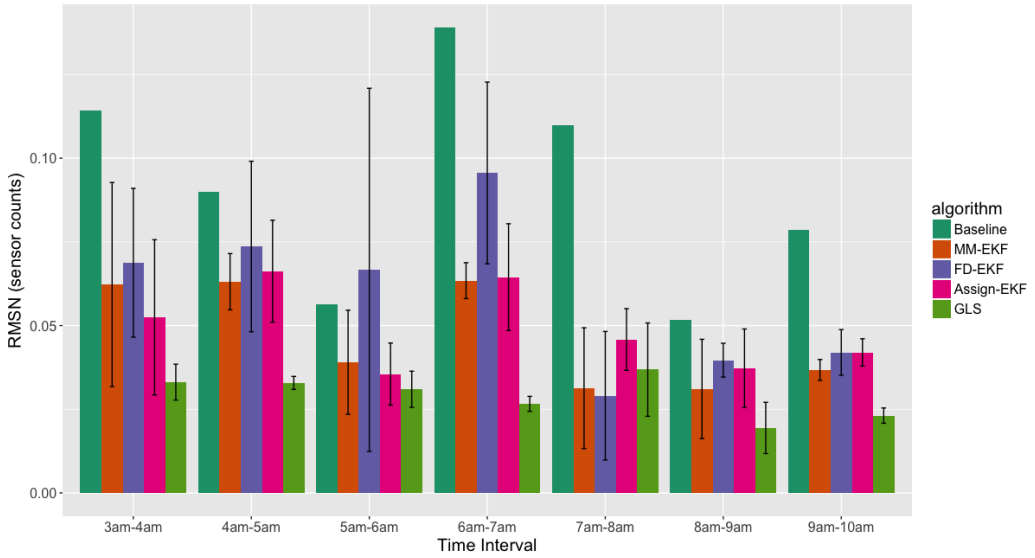

Figure 4-13: RMSN error metrics by hour for estimated sensor counts for Florian case study

of the algorithm. Error bars signifying 95% confidence intervals were calculated for the averaged hourly RMSN values among each three-run sample.

As noted in the previous section, the primary goal of online calibration is to fit the observed sensor counts. Figure 4-13 gives a bar graph for RMSN error metrics of the estimated sensor counts compared to field measurements per hour. Each grouping of bars is an hour of the simulation period as labeled on the x-axis; the MM-EKF algorithm (orange bar) is compared to the four benchmarks and 95% confidence intervals are reported. The overall performance results still hold; all algorithms outperform the baseline, a strong argument for the necessity of online calibration in adjusting to real-time observations. Of the online calibration algorithms, GLS has the smallest average RMSN values for the majority of hours and shows consistent performance below 5% RMSN across all hours of the simulation period. For the hours of 4am to 5am, 6am to 7am, and 9am to 10am in the simulation period, the average RMSN error for the GLS benchmark is significantly smaller than the average RMSN error for each of the EKF algorithms, particularly FD-EKF, as seen by the 95% confidence interval error bars. A set of pairwise Mann-Whitney statistical tests confirms this. Among the EKF algorithms, the FD-EKF algorithm generates wide confidence in-

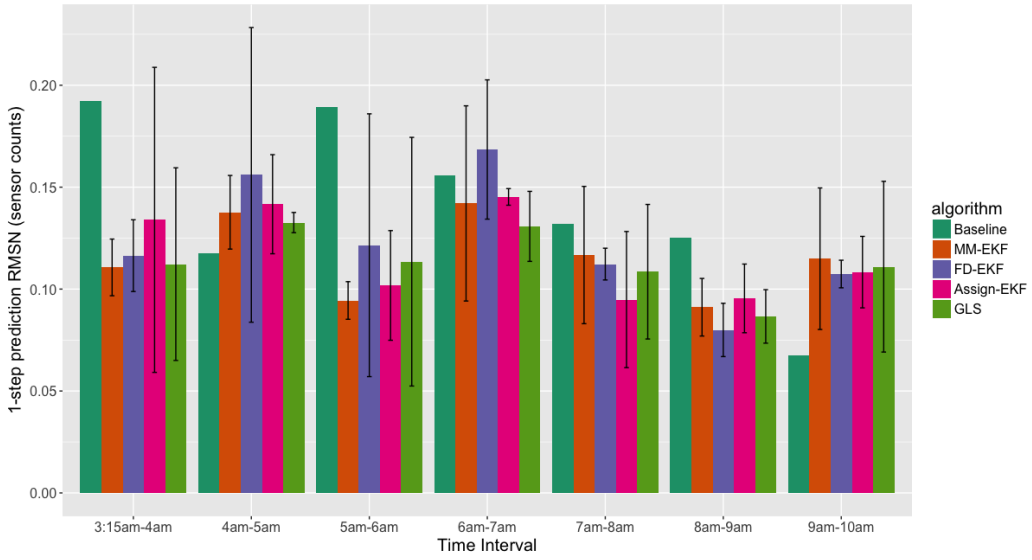

Figure 4-14: RMSN error metrics by hour for 1-step predicted sensor counts for Florian case study

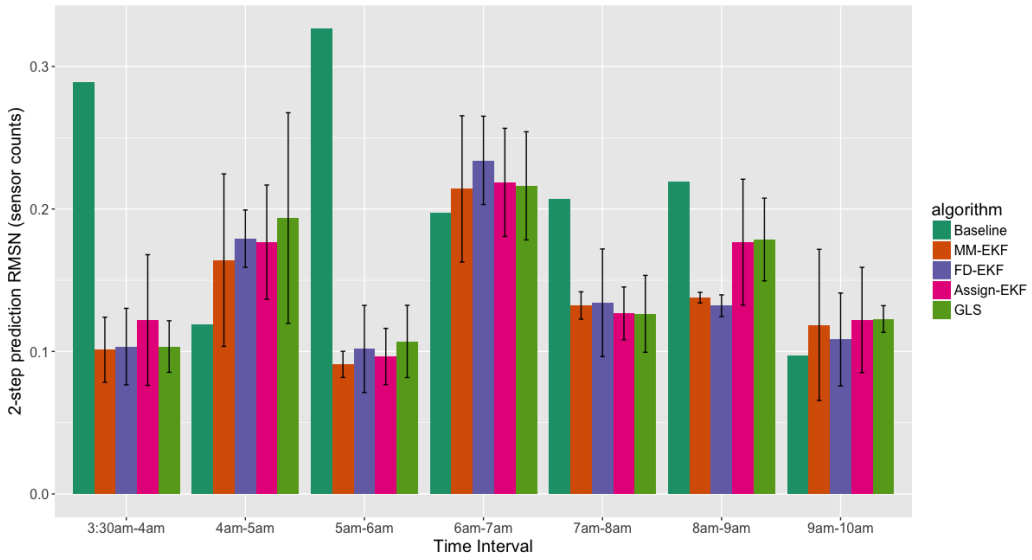

Figure 4-15: RMSN error metrics by hour for 2-step predicted sensor counts for Florian case study

tervals especially in the first half of the simulation period, indicating considerable variability among experiment runs. The Assign-EKF and MM-EKF algorithms perform comparatively across all hours of the simulation period, and both have their smallest RMSN errors in the hours after the morning peak period.

Figures 4-14 and 4-15 show the same hourly metrics for fit to sensor counts for the 1-step prediction and 2-step prediction stages. Note that the first set of bars in each figure are for the period 3:15am to 4:00am and 3:30am to 4:00am, respectively, since 1-step and 2-step predictions are not available for the periods 3:00am to 3:15am and 3:00am to 3:30am, respectively. The RMSN values per hour are more closely clustered for prediction among the different calibration algorithms than for estimation. In particular, the observation that the GLS algorithm has more similar performance to the EKF algorithms in prediction of sensor counts compared to estimation still holds in the hourly breakdown. In the hours of 4am to 5am and 9am to 10am, the baseline of prior O-D demand parameters produces small values for RMSN in 1-step and 2-step prediction. This is due to the specific shape of the synthetic demand profile used. In both of these hours, the demand profile is relatively flat so O-D parameters from the previous time interval (for 1-step prediction) or the time interval before that (for 2-step prediction) prove to be a good prediction. While the online calibration algorithms do not perform better than the baseline in every hour of the simulation period, the overall RMSN error metric for the entire simulation period as seen in Table 4.4 is still smaller than the baseline.

Figure 4-16 shows similar hour-by-hour performance metrics for estimated sensor densities. The layout of the figure is the same as for Figure 4-13. For the hours from 3am to 7am and from 8am to 10am, the performance metrics for all online calibration algorithms are of the same order of magnitude, and smaller on average than the baseline RMSN value. On the other hand, the overall worse performance of the Assign-EKF algorithm and particularly the GLS algorithm in estimating sensor densities (average RMSN of 13.1% for Assign-EKF and 16.9% for GLS compared to 10.9% for MM-EKF from Table 4.3) is due to their calibration performance during the hour of 7am to 8am. During this hour, which coincides with the morning peak period

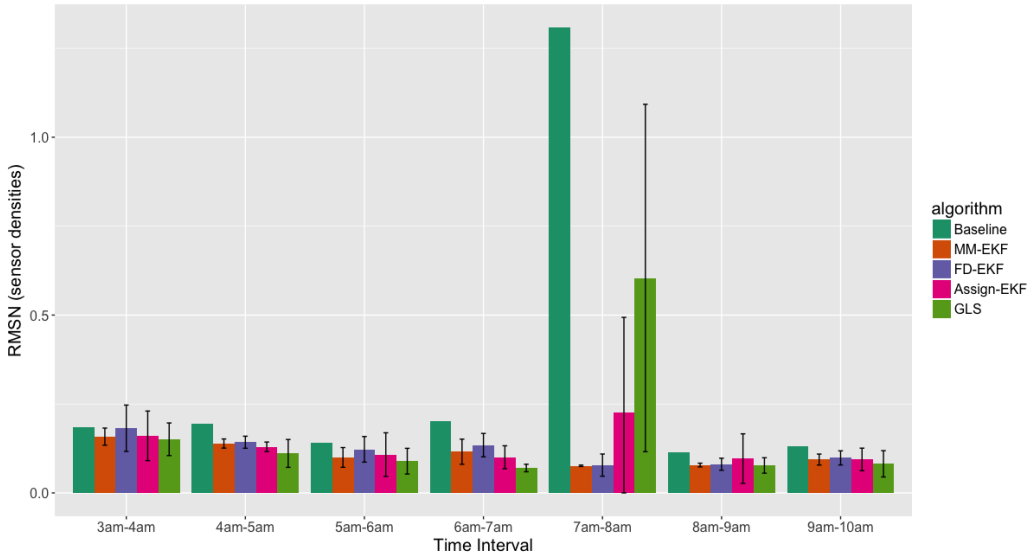

Figure 4-16: RMSN error metrics by hour for estimated sensor densities for Florian case study

demand, the synthetic prior demand shows a significant overestimation of the true demand scenario, with a total network demand in this hour of 3746 vehicles for the true demand scenario and 4029 vehicles for the prior demand scenario. This disparity is reflected in the large RMSN value for "Baseline" in estimated sensor counts from 7am to 8am in Figure 4-13, but more so in the RMSN value for "Baseline" for estimated sensor densities (RMSN of 131% from 7am to 8am). The larger prior demand leads to higher levels of congestion observed at link sensors; this can be seen in Figure 4-6. The poor performance of the baseline prior O-D demand parameters for this hour influences the calibration performance of the GLS and Assign-EKF algorithms. More specifically, both benchmark algorithms overestimate congestion during this hour by relying too much on prior parameter values, even while maintaining a close fit to sensor counts. This is reflected in the average RMSN metrics for estimated sensor densities. The GLS algorithm reports an average RMSN error of 60.4% for the hour from 7am to 8am and the Assign-EKF average RMSN is 22.7%. Likewise, the MM-EKF reports an average RMSN of 7.6% and the FD-EKF an average RMSN of 7.8% for the same hour. The difference in performance for the GLS algorithm may be due to the underlying state-space model assumption of the EKF algorithms; whereas the

GLS method has fewer constraints on the evolution of calibration parameters from one time period to the next, the parameters estimated by EKF methods are based on the transition equation of the state-space model which counterbalances the influence of the prior demand. For the Assign-EKF algorithm, the relative poor performance among EKF algorithms may be due to the empirical assignment matrix output by the simulator. As explained in Section 4.1.5, the Assign-EKF algorithm relies on the prior O-D demand parameters to generate the assignment matrix; discrepancies between the true and prior demand scenarios can lead to inaccurate gradient estimations using the Assign-EKF approach.

4.2.4 Computational performance

In addition to comparing the calibration accuracy of the different algorithms in terms of estimation and prediction of sensor count observations, analysis of the computational efficiency of the algorithms through runtime statistics is crucial to evaluating the relative performance of online calibration methods in this case study. A fundamental metric for the success of an online calibration algorithm is the ability to estimate the model parameters within the limited computational budget of one time step. While all of the algorithms run quickly due to the size of the Florian toy network, the computational results as measured by runtime (in seconds) and number of simulator evaluations still provide an important component in the assessment of the online calibration methods. Table 4.6 shows several statistics of the relative computational cost for the MM-EKF and benchmark algorithms. Each row of the table reports statistics for one algorithm run; the rows labeled "Avg" represent the average statistics across all three runs of the algorithm. The first column of metrics measures the average runtime (in seconds) of the online calibration algorithm in calculating parameter estimates for one 15-minute time interval, where the average is taken across the entire simulation period. The second column reports the portion of that runtime specifically taken to calculate the gradient. For the MM-EKF algorithm, this encompasses the "Metamodel Update," "Analytical Model Evaluation," and "Linearization" steps of Algorithm 2. For the FD-EKF algorithm, this is just the linearization step

but includes the total computation time across all simulator evaluations used to estimate the gradient. For the Assign-EKF algorithm, this just involves reading the assignment matrix from the traffic simulator, while the GLS algorithm does not require a gradient calculation. The final column of Table 4.6 gives the average number of times the traffic simulator is evaluated per 15-minute time interval, averaged across all time intervals in the simulation period. For the MM-EKF and Assign-EKF algorithms, this is just once each interval at the current state estimate. For the FD-EKF algorithm, simulator evaluations are used to numerically estimate the gradient. For the GLS, the simulator is not evaluated during the online calibration procedure.

The main takeaway from Table 4.6 is that all of the EKF methods have approximately the same average runtime (26.9 seconds for MM-EKF, 27.9 for FD-EKF, and 26.7 for Assign-EKF). As mentioned above, this is substantially less time than the computational budget of 15 minutes, given that the dimension of the problem is only 9. The similarity in computation runtimes implies that each of the gradient approximation methods also take about the same amount of time to run. As seen in the second column of the table, this is true. In fact, the portion of the runtime used for the gradient approximation step is almost negligible; most of the computational runtime of the algorithms is taken up by the other steps. For the GLS benchmark, the average runtime per interval is 10.0 seconds, which is less than half the runtime of the other algorithms. The FD-EKF benchmark algorithm also requires multiple simulator evaluations to compute the gradient approximation. On average, the FD-EKF algorithm uses 17.6 simulator evaluations per gradient approximation. Because of the small size of the network, these evaluations take a very small amount of time so the number of evaluations needed does not pose a large computational burden for the algorithm.

4.2.5 Analysis of calibrated parameters and estimated state

Since the demand scenario used in the Florian network case study is synthetic, it is possible to compare the O-D demand parameters estimated by the MM-EKF algorithm to the "true" O-D demand parameters. Figure 4-17 makes this comparison.

Algorithm	Run	Average Runtime Per Interval (sec)	Average Gradient Calculation Time Per Interval (sec)	Average Simulator Evaluations Per Interval
MM-EKF	1	22.7	1.6	1
	2	29.2	2.1	1
	3	28.9	2.1	1
	Avg	26.9	1.9	1
FD-EKF	1	24.1	1.5	17.6
	2	29.4	2.0	17.5
	3	30.1	2.0	17.7
	Avg	27.9	1.8	17.6
Assign-EKF	1	26.4	0.01	1
	2	27.1	0.01	1
	3	26.5	0.01	1
	Avg	26.7	0.01	1
GLS	1	13.9	N/A	N/A
	2	8.1	N/A	N/A
	3	8.1	N/A	N/A
	Avg	10.0	N/A	N/A

Table 4.6: Summary of computational metrics for Florian case study

The four subplots show a subset of the nine O-D pairs in the network (labeled according to Table 4.1) with non-zero demand in the true demand scenario. The x-axis spans the entire simulation period, and the y-axis plots the O-D demand for the O-D pair in vehicles per 15 minutes. The blue line shows the true O-D demand over time, the red line show the prior O-D demand, and the remaining three colors graph the three calibration runs of the MM-EKF algorithm. From this subset of O-D pairs, it is seen that the proposed MM-EKF algorithm generally estimates the correct underlying trends in the O-D demand parameters. The three lines representing the different runs are clustered closely together, which suggests a level of robustness in the face of simulator and algorithm stochasticity.

Figures 4-18 and 4-19 show the fit of the counts for each 15-minute interval (in vehicles) and densities (in vehicles per mile per lane) estimated by the MM-EKF algorithm compared to the observed field measurements. The x-axis for both figures shows the "true" values of the field measurements, while the y-axis for both figures

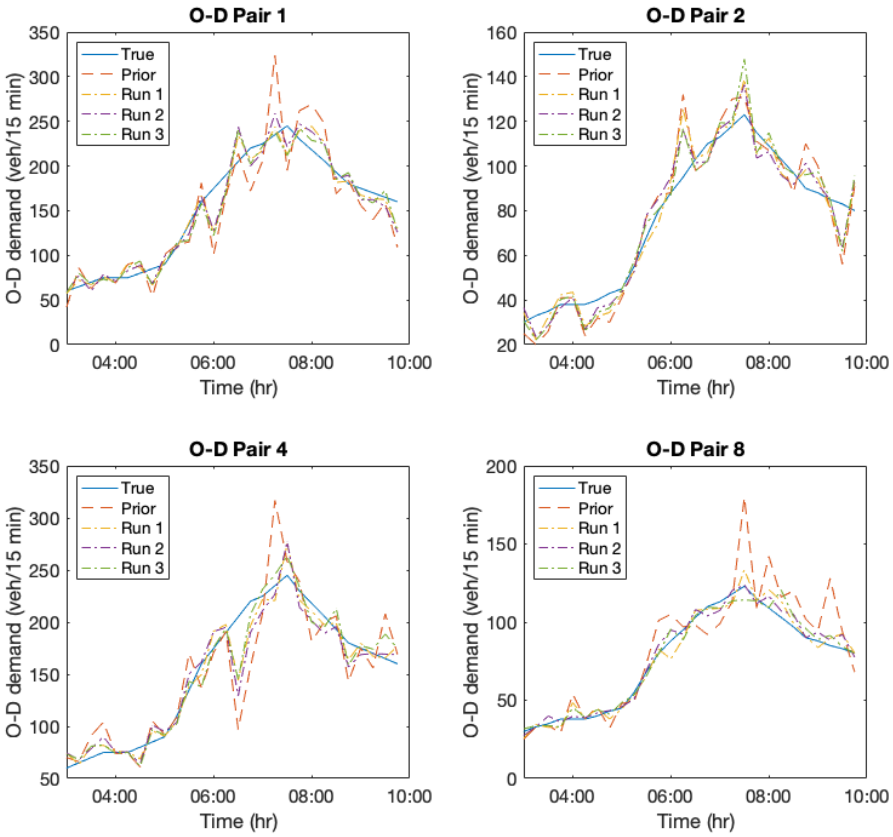

Figure 4-17: Estimated O-D demand parameters over time from MM-EKF calibration runs for Florian case study

shows the estimated values output by the MM-EKF algorithm. Each colored shape represents a separate run of the algorithm. A perfect fit, which would give a RMSN error of 0, is represented on the scatterplots by the blue "45-degree line" that corresponds to exact correlation. The fit for sensor counts seen in Figure 4-18 is very close with not a large difference among the three runs of the MM-EKF algorithm. As given in Table 4.2, the RMSN errors for the three runs are 4.6%, 4.8%, and 4.7%, indicating a very good fit. In the sensor density figure, the fit between true measurements and estimated measurements is more diffuse, and there is a higher variance for larger densities. The RMSN errors for the three runs are 11.1%, 10.6%, and 10.9%, which still indicates a strong fit to the field measurements, especially since density measurements were not used in the online calibration process. This comparison between sensor count data and sensor density data also matches what was seen in the validation of the analytical traffic model. Overall, the proposed MM-EKF algorithm is validated for a small-scale case study; it is able to successfully replicate real-time sensor count observations from the network in an online calibration setting for the Florian network case study, clearly outperforming the baseline case of no online calibration and generally maintaining similar or better performance to other benchmark algorithms in the literature.

4.3 Incorporation of multiple data sources

4.3.1 Data description

One significant advantage that the MM-EKF algorithm has over the benchmark algorithms Assign-EKF and GLS is its ability to calibrate parameters besides O-D demand parameters and incorporate other sources of field measurements than just sensor counts. Among the online calibration algorithms discussed for this case study, only the MM-EKF and FD-EKF are able to incorporate additional measurement types into their framework. To demonstrate this flexibility, another set of experiments was designed for the Florian network in which sensor densities were used as

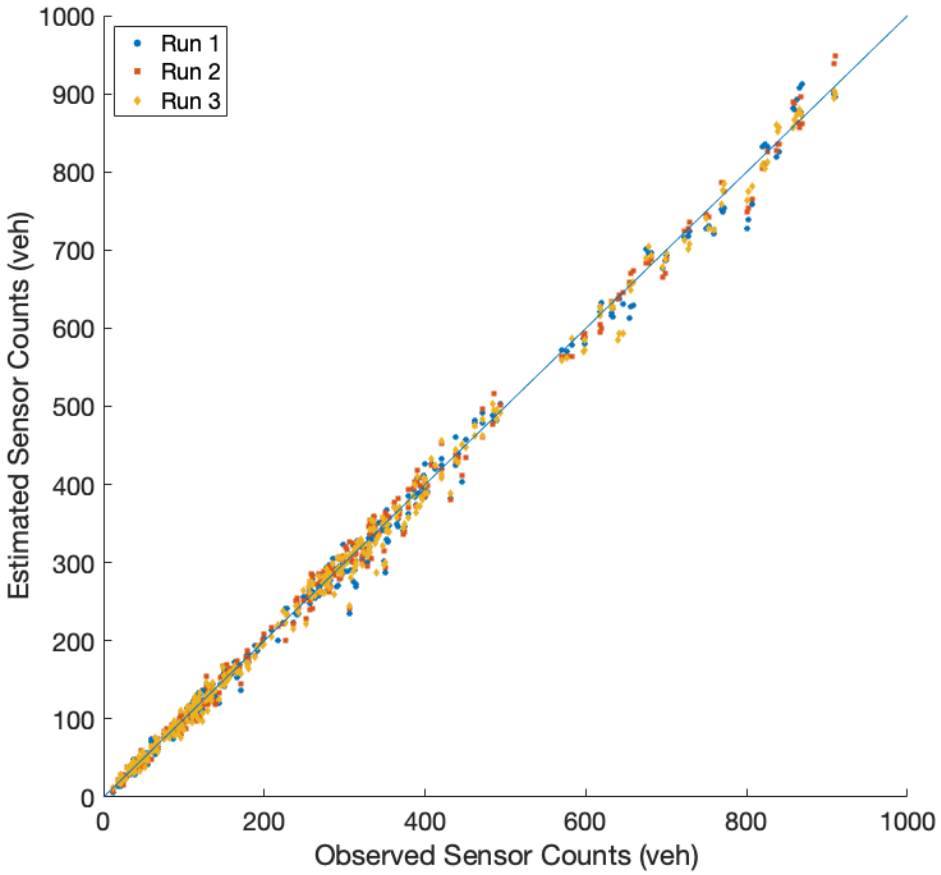

Figure 4-18: Scatterplot of field sensor counts vs. MM-EKF estimated sensor counts for Florian case study

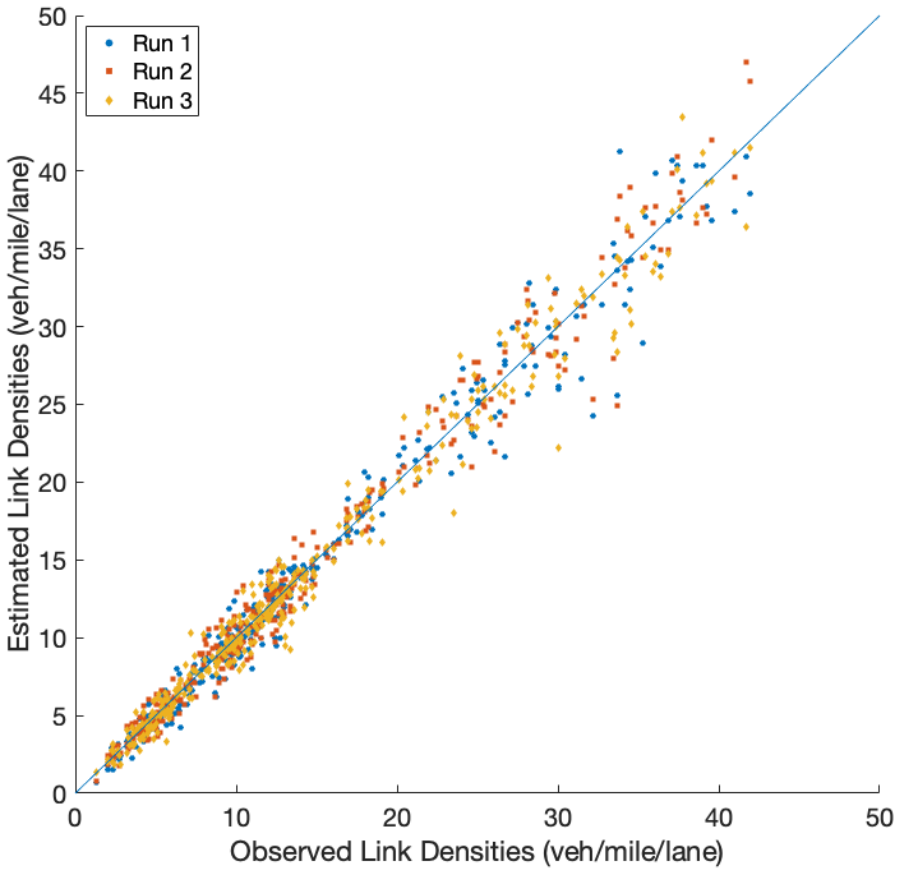

Figure 4-19: Scatterplot of field sensor densities vs. MM-EKF estimated sensor densities for Florian case study

input to the online calibration in addition to sensor counts. The proposed MM-EKF algorithm was run along with the same benchmark algorithms in order to demonstrate a twofold advantage—(1) the ability to fit both sensor count and density data under a common framework that is not possible for the Assign-EKF and GLS approaches, (2) the improved performance of the MM-EKF algorithm in estimation of sensor counts and densities compared to other flexible frameworks like FD-EKF due to the added robustness of the MM-EKF approach to simulator stochasticity provided by the analytical traffic model in the gradient estimation step.

A key concept in traffic flow theory is the idea that the same observed link counts may occur in both the uncongested regime and the congested regime due to the natural shape of the flow-density relationship. The synthetic demand profile for the new experiments takes advantage of this. The new demand profile has a simulation period of 3am to 10am, with sensor measurements (counts and densities) observed every 15 minutes. The parameters to be calibrated remain demand for the nine O-D pairs of the Florian network enumerated in Table 4.1; O-D demand is calibrated for every 15-minute time interval. For this set of experiments, only four sensors on the network are used, chosen as a subset of the twelve sensors from the previous experiment (S6, S7, S8, and S12 in Figure 4-2).

For the "true" demand scenario, the O-D demand parameters for all nine of the O-D pairs are set to the same level, and take one of two values during the simulation period. For the uncongested period of the demand profile, from 3:00am to 6:30am, each O-D pair sees a demand of 85 vehicles per 15-minute interval. Then for the congested period of the demand profile, from 6:30am to 10:00am, each O-D pair sees a demand of 140 vehicles per 15-minute interval. The demand scenario is created in this manner in order to simulate the transition from an uncongested network to a very congested network caused by a sudden onslaught of demand. In both the uncongested and congested periods, the sensors report similar link flows but the underlying network state has changed. Online calibration algorithms with a flexible framework able to accommodate multiple data sources like MM-EKF and FD-EKF can then glean additional information about the network state from further field measurements (link

densities). The synthetic "true" demand scenario is not a realistic demand profile; an immediate jump in the demand of every O-D pair in the network would not occur in a real-world setting. More specifically, the demand scenario as described violates the simple random walk assumptions of the transition equation in the state-space model formulation used for this case study (Equation 4.3). Even though the demand profile does not adhere to the modeling assumptions of the Kalman filtering approaches, it was created expressly to demonstrate the ability of the MM-EKF algorithm to fit both sensor count and density information in an online calibration setting. The proposed MM-EKF algorithm is still able to achieve good calibration results, whereas each of the benchmark algorithms fails to do the same.

For the prior demand profile, a constant setting of 85 vehicles per 15-minute interval for each O-D pair is chosen, matching the true demand profile during the uncongested period. As explained previously in Section 4.1.3, because both demand profiles are synthetic, sensor measurements must also be computed. The same process is followed. For each demand profile, sensor measurements were created using the supply-only simulator of DynaMIT by running 50 replications for the specified O-D demand and averaging the simulator outputs across all replications. Figure 4-20 shows the resulting sensor counts and densities recorded by the four sensors over the whole simulation period. Each column shows sensor measurements for one of the four sensors in the network. The first row gives sensor counts (y-axis) graphed over time (x-axis). Note that the red dashed line, which is the prior demand profile, stays relatively constant across the whole simulation period as to be expected since the O-D demand is time-independent. The blue solid line, which is the true demand profile that switches from uncongested to congested at 6:30am, registers an initial sharp increase in sensor counts as demand floods the network. Yet after the initial increase in sensor counts occurs, once congestion has propagated through the network the sensor counts settle back to values close to the uncongested sensor counts. Given just sensor counts as field measurements, it may seem as if the network has settled back to its original state. The second row of Figure 4-20 gives sensor densities (y-axis) graphed over time (x-axis). The red dashed line represents the prior demand

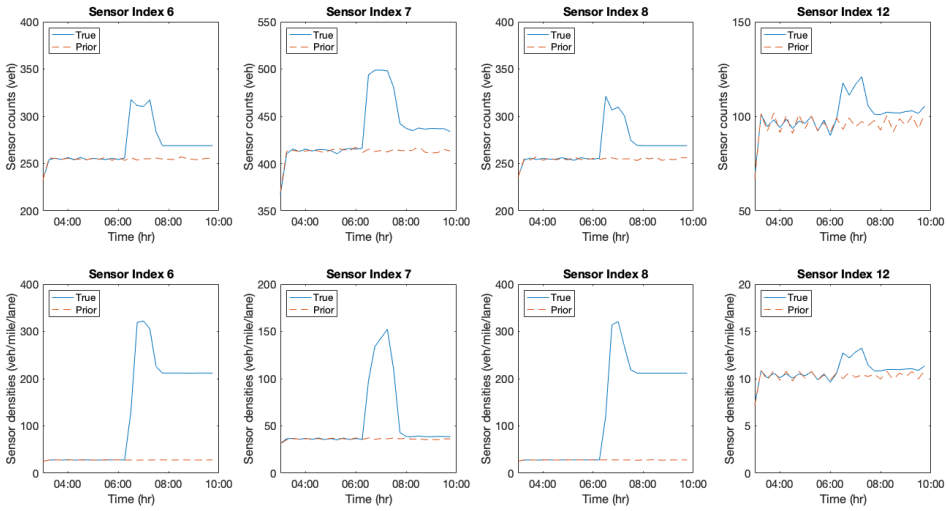

Figure 4-20: Sensor counts and densities over time for synthetic experiment on Florian network incorporating multiple data sources

profile, and the blue solid line represents the true demand profile. The impact of the additional O-D demand on the network can be observed in the density measurements at sensors 6 and 8. In the second half of the simulation period, densities for these two sensors remain near jam density, indicating congested links even with sensor counts settling back to lower numbers.

4.3.2 Experimental design

The objective of this specialized case study is to calibrate O-D demand parameters for all nine O-D pairs for every 15-minute interval in the simulation period. The majority of the experimental design is the same as for the previous experiment detailed for the Florian network in Section 4.1.4. The initialization period, algorithm parameters, underlying state-space model, and metamodel specifications remain the same. In fitting the random error parameters (a, q_0, b_1, $r_{0,1}$, b_2, $r_{0,2}$) of the state-space model offline from simulator data, the parameters for the indirect measurement equation (b_1, $r_{0,1}$) for sensor counts and sensor densities are tuned independently due to the disparity in magnitude of measurements and the difference in units. For count observations, the parameters are set to $b_1 = 0.03$ and $r_{0,1} = 1$. For density observations, the parameters

are set to $b_1 = 0.015$ and $r_{0,1} = 0.5$. In addition, the weight parameter w is set to 10 instead of 100 for this set of experiments.

The main difference in experimental design is the use of density measurements in addition to count measurements. For both the MM-EKF and FD-EKF algorithms, two versions are run—one version utilizing only count measurements, to put the algorithms on equal footing with the GLS and Assign-EKF benchmark algorithms, and one version utilizing both count and density measurements. Each algorithm is again run three times, and performance is evaluated using the root mean-square normalized (RMSN) error defined in Equation 4.5. The goal of these experiments is to determine if the general-purpose online calibration algorithms (MM-EKF, FD-EKF) can successfully incorporate the new surveillance data and provide an accurate fit to both types of measurements, a capability that the GLS and Assign-EKF algorithms lack. A secondary goal is to analyze differences in performance between the MM-EKF and FD-EKF algorithms provided both sensor count and density data due to their different methods for gradient estimation.

4.3.3 Results

Tables 4.7 and 4.8 report the performance metrics for estimation of sensor counts, sensor densities, and O-D demand parameters for all online calibration algorithms over the entire simulation period. The layout and relationship between the two tables is the same as for Tables 4.2 and 4.3. Each row of Table 4.7 shows results for one algorithm run. "Baseline" denotes the prior O-D demand profile, while the online calibration algorithms that use both count and density field data are categorized as "Counts+Densities" and those that use only count data are categorized as "Counts." The Assign-EKF and GLS algorithms can only handle count data. The RMSN error metrics reported in Table 4.7 are averaged across the entire simulation period. Table 4.8 provides a summary of the results in Table 4.7; each row shows results for the average of all three runs of each online calibration algorithm. Percent improvements are calculated off the "Baseline" row.

The main takeaway from this set of experiments is in the relative improvement

Algorithm	Run	counts		densities		O-D	
		RMSN	% improv	RMSN	% improv	RMSN	% improv
Baseline		0.070	-	0.573	-	0.196	-
MM-EKF Counts+Densities	1	0.041	41.3	0.083	85.6	0.283	-43.8
	2	0.057	18.2	0.169	70.5	0.334	-69.8
	3	0.064	8.2	0.306	46.6	0.270	-37.5
FD-EKF Counts+Densities	1	0.097	-38.7	0.373	34.9	0.392	-99.6
	2	0.115	-63.8	0.448	21.8	0.404	-105.7
	3	0.100	-41.8	0.464	19.1	0.423	-115.2
MM-EKF Counts	1	0.038	46.1	0.381	33.6	0.201	-2.1
	2	0.037	46.7	0.534	6.8	0.208	-5.9
	3	0.036	48.1	0.561	2.2	0.202	-3.0
FD-EKF Counts	1	0.032	54.9	0.552	3.8	0.202	-2.9
	2	0.033	52.9	0.552	3.8	0.208	-5.9
	3	0.037	46.7	0.558	2.6	0.215	-9.5
Assign-EKF	1	0.035	50.7	0.562	2.0	0.230	-16.9
	2	0.031	56.4	0.551	3.8	0.224	-14.0
	3	0.036	48.8	0.559	2.5	0.218	-10.7
GLS	1	0.022	68.9	0.550	4.1	0.212	-7.8
	2	0.022	68.9	0.557	2.9	0.200	-1.6
	3	0.022	68.1	0.548	4.4	0.193	1.5

Table 4.7: Estimation performance metrics for experimental results across full simulation period for Florian case study incorporating multiple data sources

Algorithm	counts		densities		O-D	
	RMSN	% improv	RMSN	% improv	RMSN	% improv
Baseline	0.070	-	0.573	-	0.196	-
MM-EKF Counts+Densities	0.054	22.6	0.186	67.6	0.295	-50.4
FD-EKF Counts+Densities	0.104	-48.1	0.429	25.3	0.406	-106.8
MM-EKF Counts	0.037	47.0	0.492	14.2	0.204	-3.7
FD-EKF Counts	0.034	51.5	0.554	3.4	0.208	-6.1
Assign-EKF	0.034	52.0	0.558	2.8	0.224	-13.9
GLS	0.022	68.7	0.552	3.8	0.202	-2.6

Table 4.8: Average RMSN error metrics for estimation across all three runs for full simulation period for Florian case study incorporating multiple data sources

in RMSN error for estimated densities for the MM-EKF Counts+Densities algorithm compared to the baseline of prior O-D demand and to all other online calibration algorithms tested. At the same time, the proposed MM-EKF Counts+Densities algorithm is able to improve fit to estimated counts compared to the baseline, finding parameter estimates that fit both count and density measurements simultaneously. The average RMSN error for fit to estimated sensor counts is 5.4%, a 22.6% improvement over the baseline, while the average RMSN error for fit to estimated sensor densities is 18.6%, a 67.6% improvement over the baseline. While on average the MM-EKF Counts+Densities algorithm improves over the baseline, the 95% confidence intervals for the two metrics for the three-run samples are $(2.5\%, 8.4\%)$ for estimated sensor counts and $(0\%, 46.6\%)$ for estimated sensor densities. The baseline RMSN for densities falls outside of the confidence interval, meaning the improvement is statistically significant, but the baseline RMSN for counts is 7.0%, which falls in the 95% confidence interval. Still, the 95% confidence interval for RMSN values for estimated sensor counts falls entirely below 10%, which has been noted to be a satisfactory fit to sensor counts as discussed in Section 4.1.5.

Among the benchmark algorithms, there are two comparisons to make with the MM-EKF Counts+Densities algorithm results. The first is to algorithms that only use counts data. For fit to estimated sensor counts, the GLS benchmark shows the greatest improvement over the baseline with a very small average RMSN error of 2.2%. This matches what was observed in Section 4.2.2. The average RMSN error for the three benchmark EKF algorithms fall between 3% and 4%. Running pairwise Mann-Whitney tests, all comparisons with the MM-EKF Counts+Densities algorithm conclude that the algorithms fitting only to counts have a better performance in estimating sensor counts than the MM-EKF Counts+Densities algorithm, with p-values smaller than 0.05. For fit to estimated sensor densities, the opposite is true. Since these benchmark algorithms only calibrate to count data, the performance metrics for fit to sensor densities is only slightly better than the baseline, as expected. The average RMSN values range from 49.2% to 55.8%. Another set of statistical tests conclude that the MM-EKF Counts+Densities algorithm performs significantly

better than these benchmarks in fit to sensor densities. A worse performance in the RMSN metric for sensor counts for the MM-EKF Counts+Densities algorithm is to be expected as new measurements (link densities) are incorporated, since the objective function of the online calibration problem must now accommodate for a minimization of a larger number of terms across both counts and densities. Still, the loss in fit to sensor counts is counterbalanced by the large improvement in fit to sensor densities.

The second comparison to make with the MM-EKF Counts+Densities algorithm is to the only other algorithm able to incorporate density observations, FD-EKF Counts+Densities. The FD-EKF Counts+Densities algorithm actually produces a worse fit to sensor counts than the baseline, with an average RMSN value of 10.4%. For fit to sensor densities, the 25.3% improvement over the baseline (average RMSN error of 42.9%) is less than that for the MM-EKF Counts+Densities algorithm, a significant difference confirmed by a Mann-Whitney test ($p < 0.05$). As the only difference between the MM-EKF and FD-EKF algorithms is in the gradient approximation in the linearization step of the EKF algorithm (all parameters are kept the same in the experimental design), the divergence in performance must lie in that step. As discussed in Section 2.3.1, the finite differences approach to gradient estimation is significantly affected by simulator stochasticity. For the MM-EKF algorithm, the use of an analytical traffic model may make the gradient calculation more robust to the effects of simulator stochasticity, leading to improved performance for the online calibration problem. This is discussed further in Section 4.3.4.

The simulation period central to analysis of the performance of the online calibration algorithms is the congested period from 6:30am to 10:00am, where the density observations provide crucial information about the network state and the prior O-D parameters differ significantly from the true O-D parameters. Tables 4.9 and 4.10 provide experiment results in the same format as Tables 4.7 and 4.8, but limited to the congested period from 6:30am to 10:00am. The takeaways are similar for the analysis of the congested period as they were for the overall simulation period, but the magnitude of the improvements are more pronounced. The proposed MM-EKF Counts+Densities algorithm shows improvement over the baseline for both fit to sen-

Algorithm	Run	counts		densities		O-D	
		RMSN	% improv	RMSN	% improv	RMSN	% improv
Baseline		0.105	-	1.068	-	0.393	-
MM-EKF Counts+Densities	1	0.050	52.3	0.093	91.3	0.502	-27.7
	2	0.075	28.6	0.254	76.2	0.532	-35.4
	3	0.093	11.6	0.537	49.8	0.486	-23.7
FD-EKF Counts+Densities	1	0.158	-50.8	0.678	36.5	0.657	-67.3
	2	0.196	-86.6	0.834	21.9	0.694	-76.7
	3	0.163	-55.5	0.855	19.9	0.733	-86.6
MM-EKF Counts	1	0.046	56.5	0.688	35.6	0.351	10.6
	2	0.043	59.1	1.004	6.0	0.352	10.4
	3	0.039	62.9	1.036	3.0	0.356	9.5
FD-EKF Counts	1	0.034	67.7	1.029	3.7	0.356	9.3
	2	0.038	63.8	1.043	2.4	0.355	9.7
	3	0.044	58.1	1.038	2.9	0.344	12.5
Assign-EKF	1	0.032	69.6	1.038	2.8	0.357	9.1
	2	0.032	70.0	1.031	3.5	0.359	8.7
	3	0.040	61.5	1.027	3.8	0.358	8.8
GLS	1	0.022	78.9	1.033	3.3	0.357	9.2
	2	0.022	79.5	1.029	3.7	0.354	9.9
	3	0.026	75.1	1.024	4.1	0.355	9.8

Table 4.9: Estimation performance metrics for experimental results for congested period for Florian case study incorporating multiple data sources

sor counts and densities. The average RMSN error for the proposed algorithm is 7.3% (30.8% improvement) for sensor counts and 29.5% (72.4% improvement) for sensor densities. While the results for density measurements are statistically significant, the baseline RMSN of 10.5% for sensor counts falls in the 95% confidence interval for MM-EKF Counts+Densities results. This is reasonable, since the "true" demand scenario was designed to show similar sensor counts in both the uncongested and congested periods, and the prior demand scenario is just the O-D demand parameters from the uncongested period. Thus, improvement over the baseline for fit to counts is not the primary metric of success. The comparative advantage of the MM-EKF Counts+Densities algorithm over the corresponding FD-EKF Counts+Densities algorithm still holds when constrained to the congested period. The FD-EKF algorithm shows an average RMSN error of 17.2% for sensor counts and 78.9% for sensor densities; both results are significantly worse than the corresponding results for the proposed algorithm as confirmed by statistical tests.

Algorithm	counts		densities		O-D	
	RMSN	% improv	RMSN	% improv	RMSN	% improv
Baseline	0.105	-	1.068	-	0.393	-
MM-EKF Counts+Densities	0.073	30.8	0.295	72.4	0.506	-28.9
FD-EKF Counts+Densities	0.172	-64.3	0.789	26.1	0.695	-76.9
MM-EKF Counts	0.042	59.5	0.909	14.9	0.353	10.2
FD-EKF Counts	0.039	63.2	1.036	3.0	0.352	10.5
Assign-EKF	0.035	67.0	1.032	3.4	0.358	8.9
GLS	0.023	77.8	1.029	3.7	0.355	9.6

Table 4.10: Average RMSN error metrics for estimation across all three runs for congested period for Florian case study incorporating multiple data sources

Figures 4-21 through 4-23 show the experimental results graphically. Figure 4-21 summarizes the performance metrics for sensor counts for the different online calibration algorithms for the whole simulation period (3:00am to 10:00am), the uncongested "control" period (3:00am to 6:30am), and the congested period (6:30am to 10:00am). Each bar represents the average RMSN error across the three independent runs, and the error bars give the 95% confidence interval from the three-run sample. For all of the confidence intervals, a sample size of only three runs leads to wide ranges that more experiments would alleviate. The uncongested regime provides a general ground truth since the prior demand is the same as true demand in this period, and predictably all algorithms do very well with RMSN values below 5%. The figure shows the same conclusions as were discussed above.

Overall, GLS has the best performance in fit to counts but all algorithms improve over the baseline for the congested period except for the FD-EKF Counts+Densities algorithm. In fact, the baseline RMSN falls below the 95% confidence interval of the FD-EKF Counts+Densities algorithm, indicating that performance shows significant deterioration compared to the baseline. This may partially be due to a need for better tuning of EKF algorithm parameters, but the impact of simulator stochasticity is also a factor to be discussed in Section 4.3.4. The wide confidence intervals for both the MM-EKF and FD-EKF approaches in the congested regime also point to the

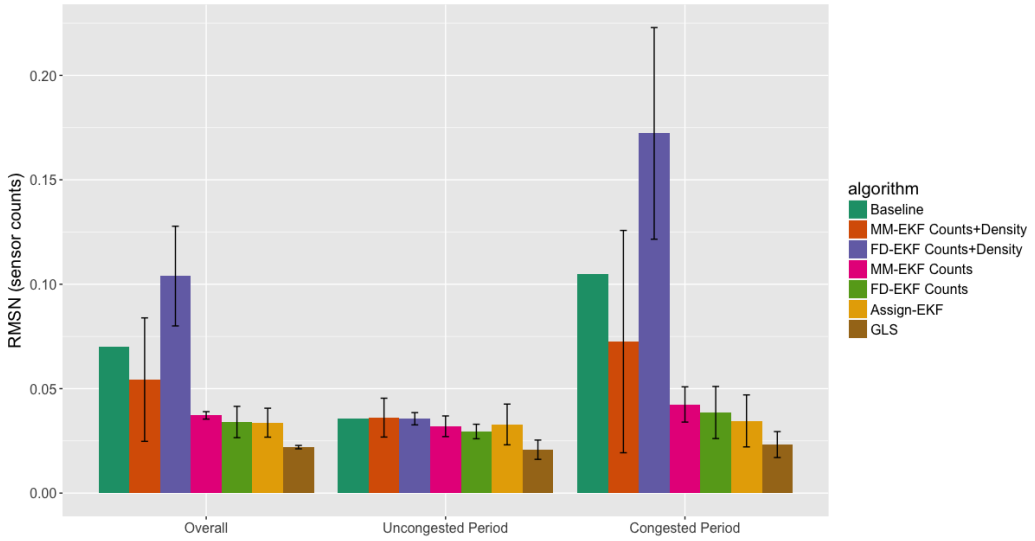

Figure 4-21: Average performance metrics for estimated sensor count measurements for MM-EKF algorithm and benchmarks for Florian case study incorporating multiple data sources

variability among the limited number of experiment runs.

Figure 4-22 presents the same performance metrics, but for fit to sensor densities. Again the uncongested regime provides a general ground truth with all algorithms performing very well with RMSN values below 10%. Overall, the MM-EKF Counts+Densities algorithm performs better than the baseline as well as the online calibration algorithms fit only to count measurements. For the congested regime, after the algorithms adjust to the initial spike in sensor count and density measurements, the benchmark algorithms fitting only to counts perform similarly to the baseline of prior O-D demand. The MM-EKF Counts algorithm does see some improvement in average RMSN in the congested period, perhaps by finding a different set of estimated O-D parameters with more similar density observations to the true scenario. The FD-EKF Counts+Densities algorithm performs better than the baseline but does not compare to the performance of the MM-EKF Counts+Densities algorithm.

Finally, Figure 4-23 presents one last set of performance metrics for fit to the underlying estimated O-D demand parameters. These metrics are also reported in Tables 4.7 and 4.9. Interestingly, the two "Counts+Densities" algorithms do not estimate O-D demand parameters similar to the true demand parameters, even though

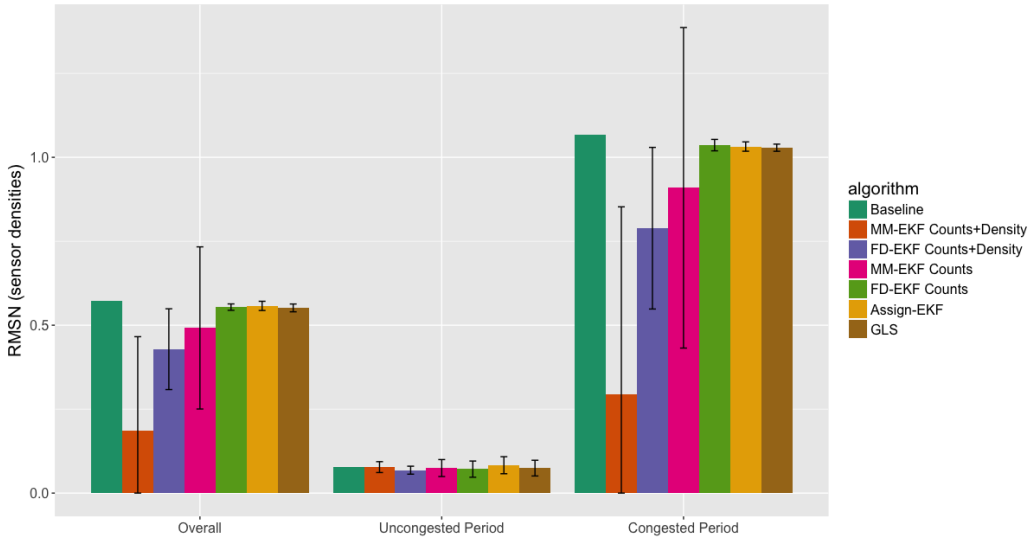

Figure 4-22: Average performance metrics for estimated sensor density measurements for MM-EKF algorithm and benchmarks for Florian case study incorporating multiple data sources

the MM-EKF Counts+Densities algorithm is able to fit both sensor counts and densities well. The average RMSN error for fit to O-D demand for the MM-EKF Counts+Densities algorithm is 29.5%, compared to a RMSN error of 19.6% for the baseline of prior O-D demand. The plotted 95% confidence intervals show that for this metric, all of the online calibration algorithms given only sensor count measurements demonstrate improved performance over the baseline during the congested period. The wide confidence intervals for the "Counts+Densities" algorithms imply a large variation in estimated O-D parameters across the three experiment runs, suggesting that in the aim of calibrating both counts and densities, the algorithm runs show divergence in the underlying states estimated by the Kalman filter. While fit to the "true" O-D parameters is not the explicit objective of the online calibration problem, accurately estimating the underlying state is a secondary goal. The poor performance presented here possibly arises from the imprecision of the state-space model assumptions, particularly the simple random walk transition equation, mentioned in Section 4.3.1.

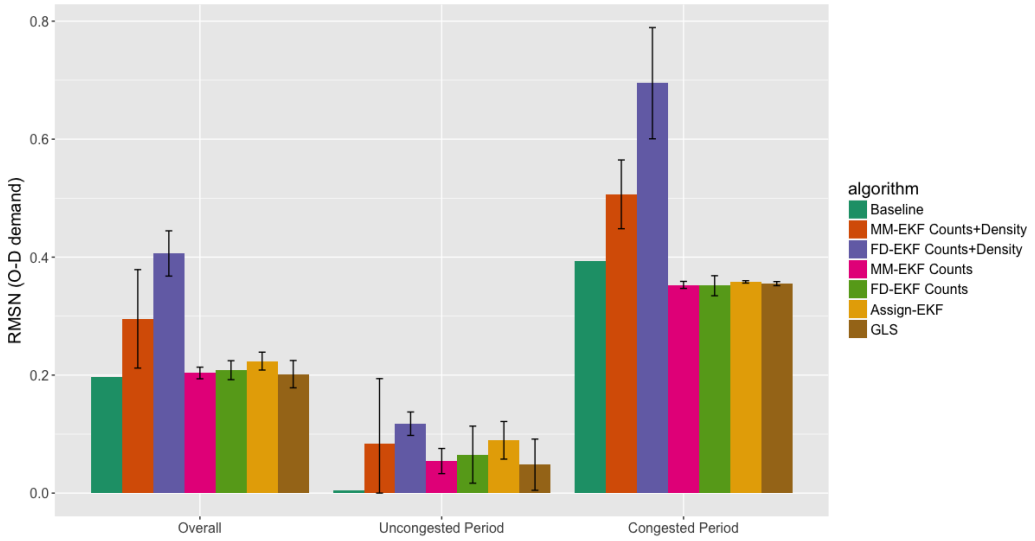

Figure 4-23: Average performance metrics for estimated O-D demand parameters for MM-EKF algorithm and benchmarks for Florian case study incorporating multiple data sources

4.3.4 Simulator stochasticity and gradient estimation

The results of this case study point to a sharp difference in performance between the proposed MM-EKF algorithm and the FD-EKF benchmark algorithm, even though the only difference between the algorithms is in the gradient estimation methods of the linearization step of the EKF algorithm. For the FD-EKF benchmark, estimating the gradient of the stochastic traffic simulator using central finite differences implies a trade-off between increased accuracy from methodological adjustments to control for simulator stochasticity and increased computational efficiency of minimizing the number of simulator evaluations required during each time step of the online calibration process. There are many factors that play into the accuracy of the central finite differences approximation of the gradient, from the step size of the perturbation to the number of simulator replications averaged at each perturbation. Since the accuracy of the gradient estimate plays a significant role in the performance of the Kalman filter, this decision of how to systematically address the trade-off between noise stemming from simulator stochasticity and a limited computational budget is crucial for practitioners to answer in implementation of the FD-EKF algorithm. These considerations

are especially important when applying FD-EKF to a high-dimensional problem as computational costs grow linearly with problem size.

Simulator stochasticity also plays a role in the gradient estimation step of the proposed MM-EKF algorithm, but only through the generation of simulator-evaluated points used to estimate the metamodel parameters as discussed in Section 3.2. For simulator-evaluated points generated offline, the additional computational cost of running multiple simulator replications and averaging the results to decrease the effect of simulator stochasticity does not impact the limited computational budget in the online phase. For those generated online, some consideration does need to be given to the effect of simulator stochasticity on the simulator outputs for the sampled points used to fit the metamodel. Yet the simulator points are only used to estimate the metamodel parameters and not directly for the calculation of the gradient. Once the simulator-evaluated points have been generated, the gradient estimation step of the MM-EKF algorithm is entirely analytical and not impacted by simulator stochasticity. In addition, the metamodel formulation is such that it relies on the use of problem-specific structural information from a macroscopic traffic model; this global structural information can provide a counterbalance to the stochasticity of the simulator.

Though some methodological adjustments have been proposed to control for simulator stochasticity in the context of gradient estimation (for instance, masking the H matrix with prior knowledge about network structure [73]), a straightforward approach to minimize stochasticity is to average the estimation results from multiple replications of the simulator. This approach is tested for the above experiment, but first the impact of simulator stochasticity on the gradient approximation in the FD-EKF algorithm is illustrated with a simple example.

For one of the runs of the FD-EKF Counts+Densities algorithm presented in Section 4.3.3, the gradient of the simulator during the calibration interval from 3:15am to 3:30am, denoted as $\mathbf{H}$, is estimated using 50 separate replications. That is, 50 different estimates of the simulator gradient at the same point are produced. The gradient $\mathbf{H}$ is an 8x9 matrix since the online calibration problem uses eight field measurements (four sensor counts and four sensor densities) to calibrate nine O-D

Quantile	Min	25%	50%	75%	Max
CV	-447.61	0.40	1.96	5.88	107.83

Table 4.11: Distribution of coefficients of variation for 50 replications of FD-EKF gradient estimate

demand parameters. To quantify the variation among the different gradient estimates, the coefficient of variation (defined as the standard deviation divided by the mean) is calculated for each of the 72 elements of the gradient. The quantiles of the 72 coefficients of variation (CV) are listed in Table 4.11. As seen in the table, the coefficients of variation span a very wide range with some significant outliers. The coefficient of variation is a standardized metric of dispersion; the median CV of the elements of the gradient estimates being equal to 1.96 suggests significant variability among the 50 replications of the same simulator gradient. While this exploratory analysis is only one example of the calculation of the gradient estimate, it does indicate that addressing simulator stochasticity is a major consideration for the FD-EKF algorithm. More work on the topic of quantifying simulator stochasticity can been found in the literature along with guidance on how best to control for simulator stochasticity, particularly in the work of Zhang [73].

For the experimental design laid out in Section 4.3.2, one method to minimize simulator error using multiple replications averaged for the gradient estimate is pursued. For the case study, the FD-EKF Counts+Densities algorithm is run using 1 replication (same results as in Section 4.3.3), 10 replications, and 50 replications averaged to estimate the gradient at each linearization step of the EKF algorithm. Each variation of the FD-EKF Counts+Densities algorithm is run three separate times.

Figures 4-24 through 4-26 show the results of this analysis. Figure 4-24 shows average RMSN errors for sensor counts during the entire simulation period (3:00am to 10:00am), the uncongested period (3:00am to 6:30am), and the congested period (6:30am to 10:00am). Within each grouping, the three variations of the FD-EKF algorithm are compared to the baseline of prior O-D demand, and to the MM-EKF results from Section 4.3.3. The error bars show 95% confidence intervals as calculated across the three algorithm runs. The average RMSN value for estimation of sensor counts

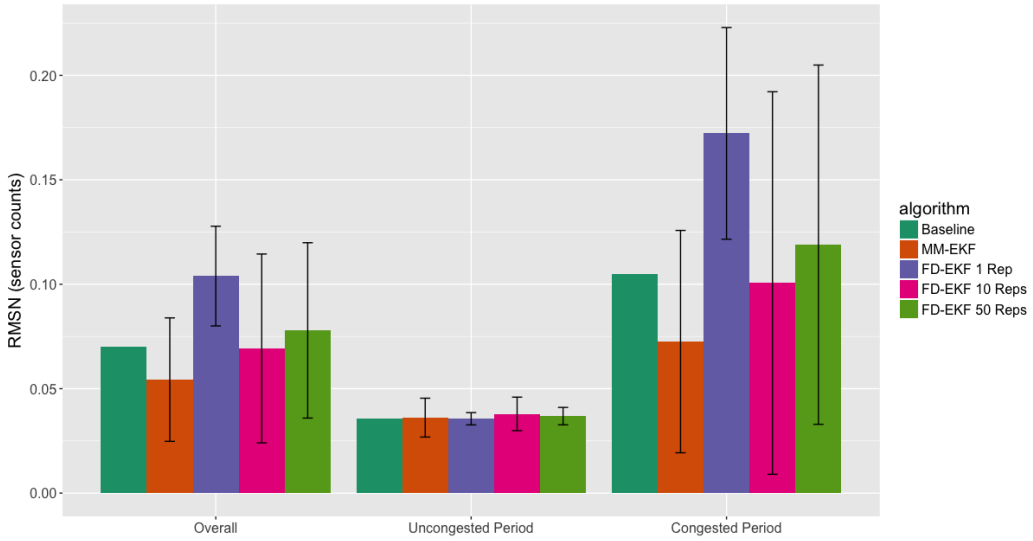

Figure 4-24: Average performance metrics for estimated sensor count measurements for gradient estimation experiments for FD-EKF algorithm

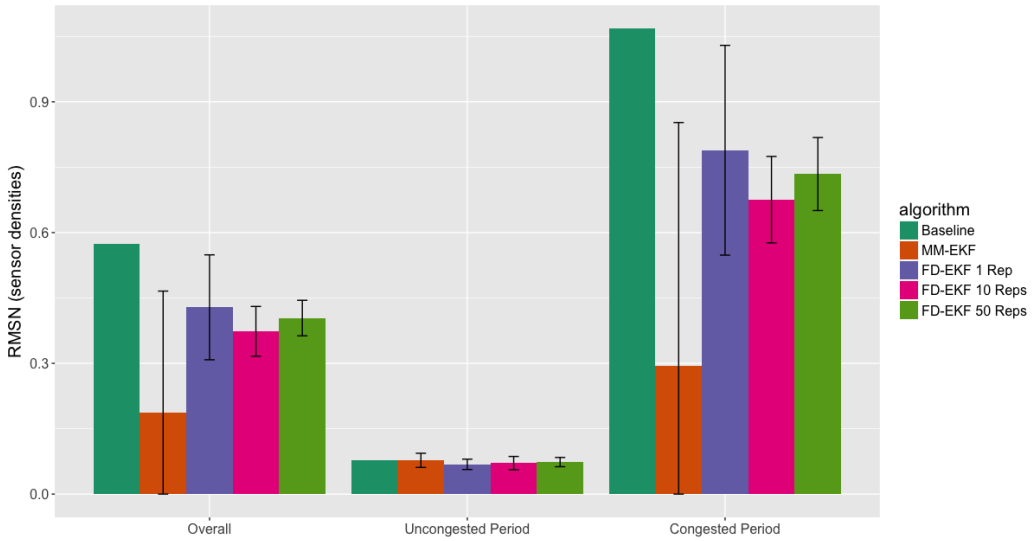

Figure 4-25: Average performance metrics for estimated sensor density measurements for gradient estimation experiments for FD-EKF algorithm

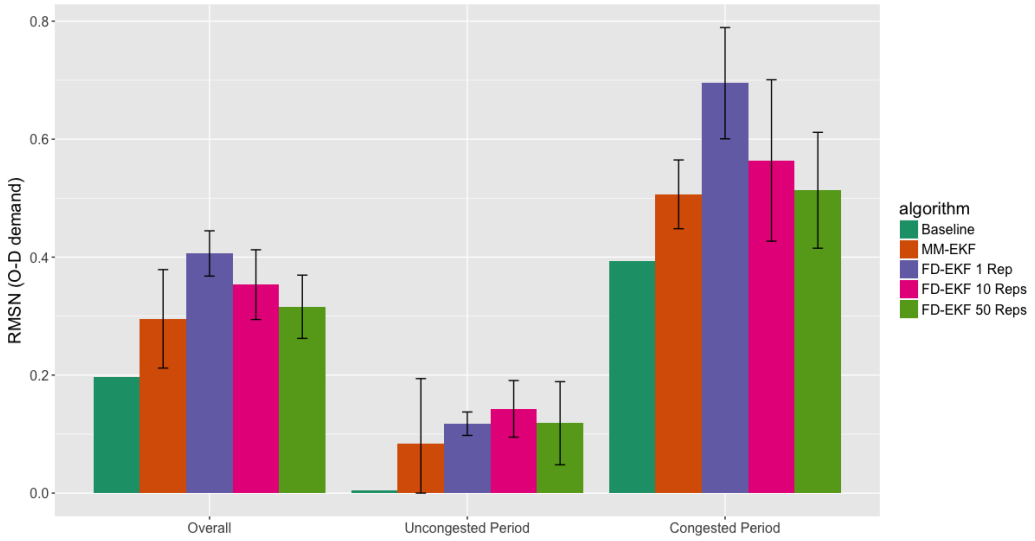

Figure 4-26: Average performance metrics for estimated O-D demand parameters for gradient estimation experiments for FD-EKF algorithm

is 10.4% for the FD-EKF algorithm using 1 replication, 6.9% for the version using 10 replications, and 7.8% for the version using 50 replications. The 95% confidence intervals for the FD-EKF versions are wide, which suggests sizable variation among runs, but on average the RMSN metric shows improvement in fit to sensor counts with more replications in estimating the gradient. One-sided Mann-Whitney tests confirm that the improvement in performance for both the 10 replications version and 50 replications version are statistically significant. The difference in performance between using 10 and 50 replications to estimate the gradient in terms of fit to sensor counts is statistically insignificant.

Figure 4-25 has the same format as Figure 4-24 but shows results for the fit to sensor densities. There is little difference among the FD-EKF versions in regard to this performance metric. Pairwise Mann-Whitney tests among the three FD-EKF versions are all statistically insignificant. Simulator stochasticity has been shown to have varying effects on different measurement outputs [73], and coupled with incorrect tuning of the algorithm parameters, this might lead to the inconclusive results seen here. Finally, Figure 4-26 shows results for the fit to the underlying O-D demand parameters. Mann-Whitney tests show the improvement of the 10 replications

version and 50 replications version over the 1 replication version in fitting O-D demand parameters ($p < 0.05$). The difference in performance between using 10 and 50 replications is statistically insignificant. The 95% confidence intervals in the figure also corroborate the benefit of additional replications in estimating the gradient, seen especially for the congested period.

Although increasing the number of replications used to estimate the gradient for the FD-EKF method leads to improved performance in terms of average RMSN error metrics, it also increases the runtime of the algorithm. While each algorithm run for this case study on the Florian toy network is able to estimate the next set of parameters well within each 15-minute time interval, the relative computational costs as measured by runtime (in seconds) and number of simulator evaluations still provide a sense of the impact of using multiple replications to improve the accuracy of gradient approximations. Table 4.12 shows several statistics of the relative computational cost for the MM-EKF and FD-EKF Counts+Densities algorithms. The set up is similar to Table 4.6. Each row of the table reports statistics for one algorithm run; the four algorithms shown in the table are the same four algorithms plotted in Figures 4-24 through 4-26. The first column of metrics measures the average runtime (in seconds) of the online calibration algorithm in calculating parameter estimates for one 15-minute time interval of the simulation period. The second column reports the portion of that runtime specifically taken to calculate the gradient. For the MM-EKF algorithm, this encompasses the "Metamodel Update," "Analytical Model Evaluation," and "Linearization" steps of Algorithm 2. For the FD-EKF algorithms, this is just the linearization step but includes the computation time for across all replications used to estimate the gradient. The final column of Table 4.12 gives the average number of times the traffic simulator is evaluated per 15-minute time interval. For the MM-EKF algorithm, this is just once each interval at the current state estimate. For the FD-EKF algorithms, simulator evaluations are used to numerically estimate the gradient—the number of evaluations scales linearly with the number of replications.

As expected, increasing the number of replications used by the FD-EKF algorithm

to approximate the gradient also increases each of the computational cost metrics as well. For the FD-EKF version using only one replication, the metrics are on the same order of magnitude as for the MM-EKF algorithm. The average runtime per interval across all three runs is 29.1 seconds for MM-EKF and 32.1 seconds for FD-EKF 1 Rep; similarly, the average runtime for the gradient calculation step is 2.1 seconds for MM-EKF and 4.0 seconds for FD-EKF 1 Rep. The majority of the runtime for both of these algorithms is taken up by the other steps of the EKF algorithm, and both gradient calculation methods are relatively fast for the small toy network. As the number of replications increases for the FD-EKF algorithm, gradient calculation takes up a larger proportion of the total runtime; the other EKF algorithm steps are essentially a fixed computational cost per time interval calibrated. Even for a low-dimensional case study, running the FD-EKF algorithm with 50 replications takes on average 112 seconds per 15-minute time interval, though the runtimes reported are for running the replications sequentially. While increased parallelization and even hardware enhancements can improve this metric, this added computational cost for the sake of minimizing simulator stochasticity in the gradient estimation step will be a significant limitation in the implementation of the FD-EKF algorithm on a large-scale network, as seen in Section 5.2.4.

These experimental results support using the proposed MM-EKF algorithm to incorporate multiple sources of field measurements in calibration of the simulator over the FD-EKF benchmark, especially in situations where the ability or budget to tune algorithm parameters is limited. As the availability of real-time data from urban road network expands, this added functionality is important for providing accurate estimation of the traffic state for prediction and guidance capabilities. Also, in situations that experience large variations in demand or are prone to significant fluctuations in driver behavior or environmental circumstances (e.g., weather conditions), the use of more types of field data than just sensor counts is crucial in delineating the true state of the traffic network. For these situations, a more flexible online calibration framework like MM-EKF that is able to find parameters that fit multiple sources of data is advantageous to benchmarks like GLS.

Algorithm	Run	Average Runtime Per Interval (sec)	Average Gradient Calculation Time Per Interval (sec)	Average Simulator Evaluations Per Interval
MM-EKF	1	26.3	1.8	1
	2	31.5	2.4	1
	3	29.5	2.1	1
	Avg	29.1	2.1	1
FD-EKF 1 Rep	1	30.6	3.8	18.2
	2	37.1	5.9	18.1
	3	28.7	2.4	18.3
	Avg	32.1	4.0	18.2
FD-EKF 10 Reps	1	45.0	19.5	175
	2	44.9	19.8	179
	3	45.4	20.2	174
	Avg	45.1	19.8	176
FD-EKF 50 Reps	1	106	81.4	877
	2	109	84.2	888
	3	121	95.7	888
	Avg	112	87.1	884

Table 4.12: Summary of computational metrics for gradient estimation experiments for MM-EKF and FD-EKF algorithms for Florian case study

4.4 Conclusion

The proposed algorithm and relevant benchmarks were run for a simple demand scenario on a toy network. The relative accuracy of the approach was measured for estimation, 1-step prediction, and 2-step prediction of sensor counts. All online calibration algorithms performed better than the baseline of offline-estimated prior information for all metrics, confirming the importance of online calibration to adjust parameters for real-time field measurements. In addition, the reported RMSN errors for the MM-EKF were small, lower than 15% for both estimation and prediction of sensor counts. RMSN errors for estimated O-D parameters and sensor densities, which are both auxiliary performance metrics, were also lower than 15%. This set of experiments validates the proposed approach as a viable algorithm for online calibration.

In the analysis on the Florian network case study, the algorithm performance on a few demand settings was able to draw out the relative strengths of the proposed Metamodel EKF algorithm, especially relative to widely-used online calibration algorithms like generalized least-squares for the real-time OD calibration problem. The general algorithm framework described in the previous chapters was successfully incorporated into a specific online calibration problem context, for a particular Dynamic Traffic Assignment system. Results showed comparable performance to various benchmarks, and improvement in its ability to incorporate multiple data sources. The following chapter shows the performance of the proposed algorithm for a high-dimensional calibration problem on a large-scale network, and the results and challenges are discussed.

Chapter 5

Singapore Expressway Case Study

The Florian network case study demonstrated the performance of the proposed Metamodel EKF (MM-EKF) algorithm on a small toy network, and highlighted advantageous characteristics of the algorithm compared to benchmark algorithms in the field of online calibration. In this chapter, the feasibility of the Metamodel EKF algorithm is evaluated on the Singapore expressway network in order to show an application of the algorithm in a large-scale case study. The presented online O-D demand calibration problem on the Singapore network is one of largest seen in the current literature, as detailed in Section 1.3. The performance of the approach is measured against suitable benchmarks, the feasibility of the method in terms of computational cost in a real-time context is shown, and practical considerations for the scalability and deployment of the algorithm are discussed.

The case study in this chapter addresses a few research questions. First, the analytical traffic model presented in Section 3.3 is validated on a high-dimensional and time-dependent demand scenario. The model is able to accurately reproduce link characteristics output by the stochastic traffic simulator for a set of transient origin-destination (O-D) demand matrices on the large-scale Singapore expressway network, and the analytical model solution is found in a matter of seconds. Second, and most importantly, the performance of the proposed Metamodel EKF algorithm is evaluated in a real-world setting and compared to several benchmark online calibration methods. It performs favorably against black-box Kalman filter methods in terms

of ease of implementation, performance accuracy, and computational cost. The empirical results validate that the proposed algorithm can be implemented in real-time for a problem of dimension 4050 and produce accurate estimates of the traffic state. Third, algorithmic considerations detailed in the previous chapters are explored in a real-world context, leading to initial empirical observations of the importance of algorithm development in practical settings. In particular, the impact of generating a bank of simulator-evaluated points offline for fitting the metamodel is observed on performance of the algorithm in estimating model parameters.

Section 5.1 gives the specifications and experimental set-up of the case study, including a description of the Singapore expressway network, the data generation process, the experimental design, performance metrics, and benchmark algorithms. Section 5.2 presents the results of the case study in detail, starting with the validation of the analytical traffic model, followed by analysis of the online calibration results both in terms of the accuracy of fit to field measurements and the online computational cost of the proposed algorithm.

5.1 Case study specifications

As in the Florian network case study, the online calibration problem for the Singapore network case study is to calibrate time-dependent O-D demand parameters provided sensor count measurements from the field. The online calibration algorithms have been implemented with DynaMIT-R, described in Section 4.1.1.

5.1.1 Network description

The case study detailed in this chapter is run on the network of major expressways and arterials in the city of Singapore, seen in a high-level summary in Figure 5-1. A schematic of the road network representation in the DynaMIT-R traffic simulator is given in Figure 5-2. The network features 1150 links, over 2300 lanes, 924 total nodes, 127 origin nodes, and 141 destination nodes. This leads to a total of 4050 O-D pairs to calibrate demand for, with over 18,000 routes. There are 172 link sensors

Figure 5-1: High-level map of the Singapore expressway network [52]

spread throughout the network that provide real-time traffic information (i.e., sensor counts). For the experiments described, demand parameters are calibrated for all 4050 O-D pairs.

5.1.2 Data description

The objective of the Singapore expressway network case study is to calibrate O-D demand parameters across a typical morning peak period in time intervals of 15 minutes. The calibration parameter vector in this case study has dimension 4050, and the measurement vector has dimension 172 consisting of sensor counts for every 15-minute time interval. Due to a lack of access to real-world traffic data for this network, a synthetic demand scenario is created for the simulation period of 6:00am to 1:00pm. O-D demand is specified for every 15-minute period and mimics a general traffic pattern of morning peak period demand, with total network demand increasing until the hour from 8:00am to 9:00am and then gradually decreasing from the morning peak. Total network demand stabilizes around 10:30am after peak demand has dissipated and remains steady until the end of the simulation period at 1:00pm. The O-D demand parameters represent the number of vehicles in each 15-minute time interval

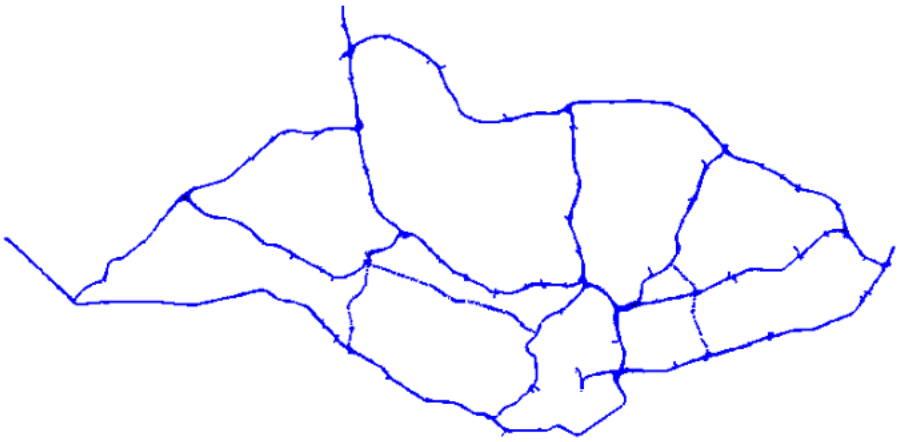

Figure 5-2: DynaMIT-R representation of the Singapore expressway network [73]

that depart from the origin node for the destination node.

In addition to the synthetic "true" demand scenario created, a set of prior O-D demand parameters is also created for the same simulation period to represent the input of offline-calibrated parameter values in the online calibration problem. To create prior demand parameter values that realistically resemble the result of an offline calibration process, the same set of O-D demand parameter values is assumed for each of four smaller time windows of equal length in the simulation period. The four time windows across which O-D demand parameters are assumed constant are 6:00am to 7:45am, 7:45am to 9:30am, 9:30am to 10:45am, and 10:45am to 1:00pm, which partition the full simulation period. This set-up for prior demand models a situation where offline calibration would provide average parameter settings over longer periods of time for the traffic network. These prior parameter values would have been calibrated using a database of historical information. For this synthetic demand scenario, the numerical values of the prior demand parameters for each O-D pair are calculated by averaging the "true" demand parameter values for that O-D pair across every time window.

Figure 5-3 plots the total demand summed across all 4050 O-D pairs (y-axis) for each 15-minute time interval in the simulation period (x-axis) for both the synthetic "true" demand profile and the synthetic "prior" demand profile. The blue solid line

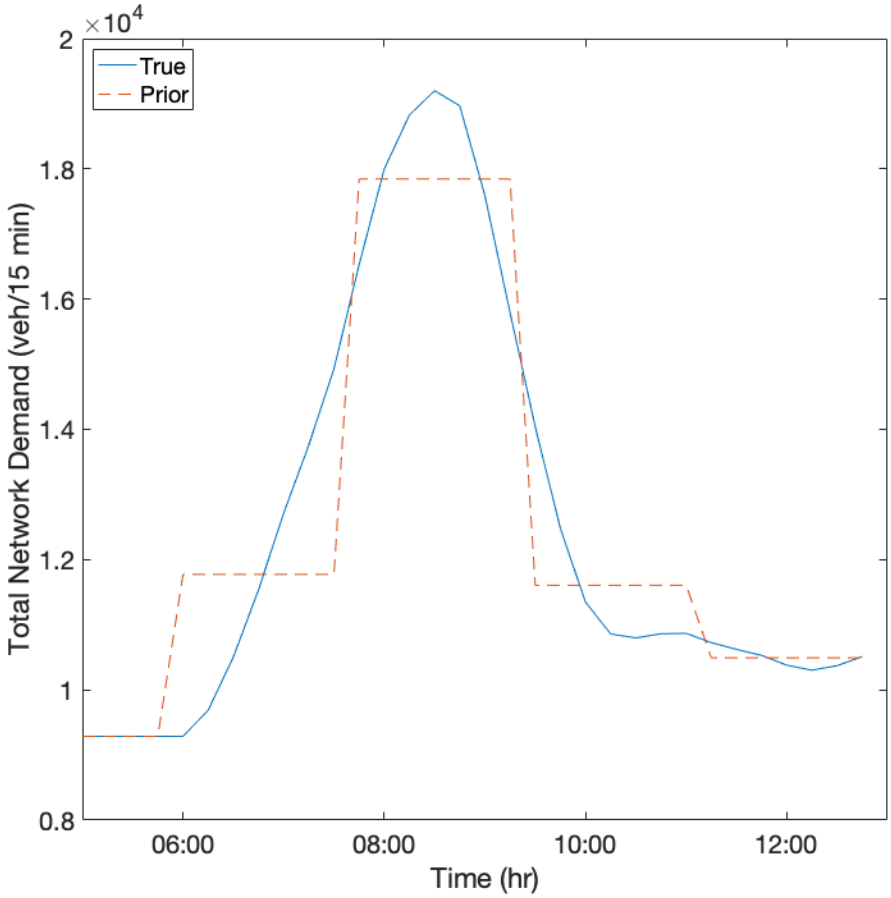

Figure 5-3: Comparison of total network O-D demand for true and prior demand scenarios for Singapore case study

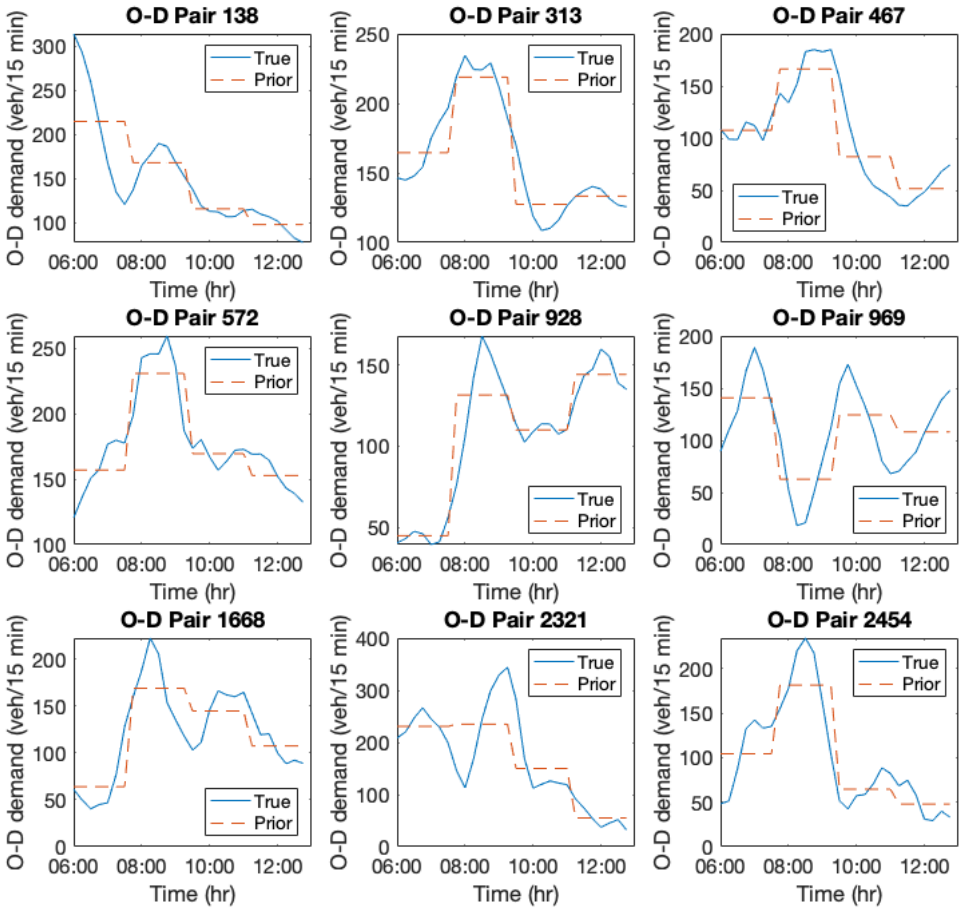

Figure 5-4: Comparison of true and prior O-D demand for subset of Singapore O-D pairs

represents the true demand, while the red dashed line represents the prior demand. The true demand curve follows the shape described above, with morning peak period demand occurring between 8:00am and 9:00am and smooth fluctuations across time. The trends for the prior demand profile are similar to those of the true demand profile, with total demand at its largest during the 7:45am to 9:30am time window that encompasses morning peak period. Notice that the prior demand is essentially a step function, with constant demand values across each of the four sub-periods as was constructed. Figure 5-4 shows a more granular view of the two demand profiles for a subset of the 4050 O-D pairs. The subset was chosen by determining the nine O-D pairs with the largest demand as summed over the entire simulation period. Each subplot shows demand (in vehicles per 15-minute interval on the y-axis) for a single O-D pair in the network across the whole simulation period. The blue solid line represents true demand, and the red dashed line represents prior demand. As can be seen, there is a lot of variation in demand profiles for the 4050 individual O-D pairs in the network across time both in magnitude and in shape. For each O-D pair, while the prior demand curve follows the same general trends as the true demand curve, it often misses the exact peaks and valleys of the true demand due to it averaging demand across the four longer time windows.

Since both the true and prior demand profiles are synthetic, field measurements must also be synthetically generated for the case study. For each demand profile, sensor measurements are created using the supply-only simulator of DynaMIT by running 50 replications for the specified O-D demand and averaging the simulator outputs across the 50 replications, just as with the Florian network case study. Through this process, sensor count measurements for the 172 link sensors are acquired and taken to be the "observed" field data.

Figure 5-5 plots count measurements (in vehicles) obtained from this process on the y-axis for each 15-minute interval on the x-axis for a subset of the 172 link sensors. The subset was chosen by determining the nine sensors with the largest count volume as summed over the entire simulation period. Each subplot graphs the measurements for a different sensor. The observed measurements for the true demand profile are

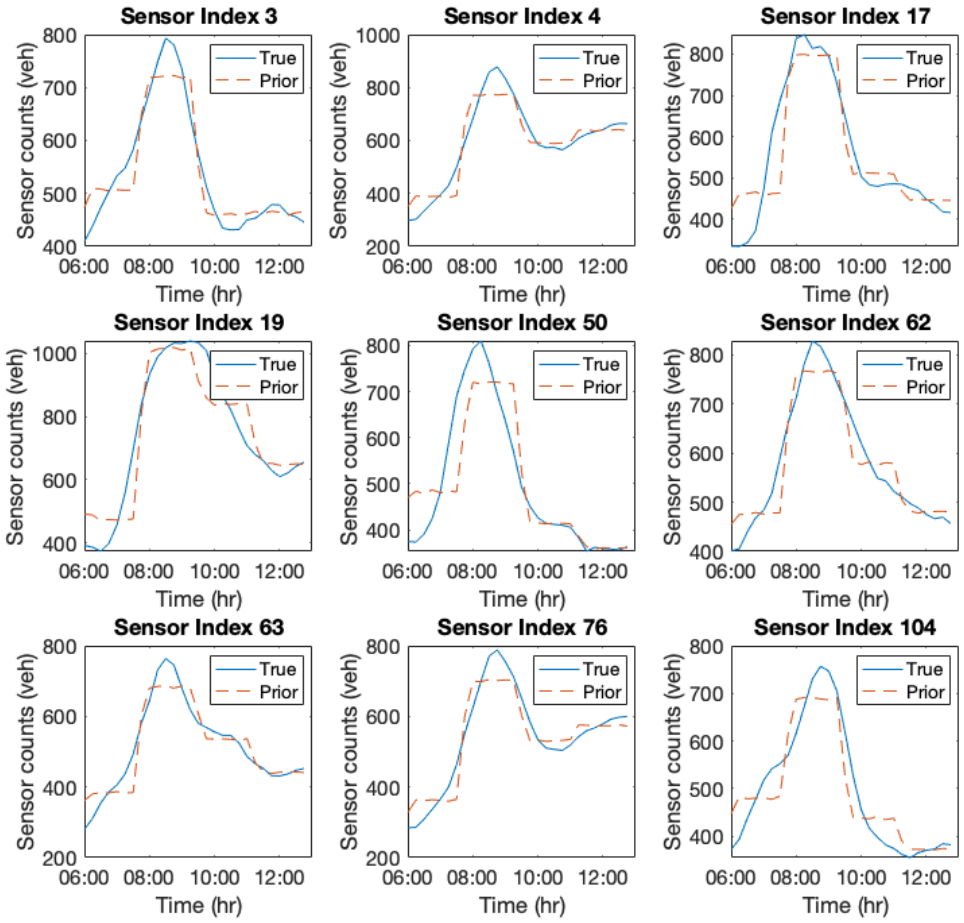

Figure 5-5: Comparison of sensor counts for true and prior demand profiles for subset of sensors in Singapore case study

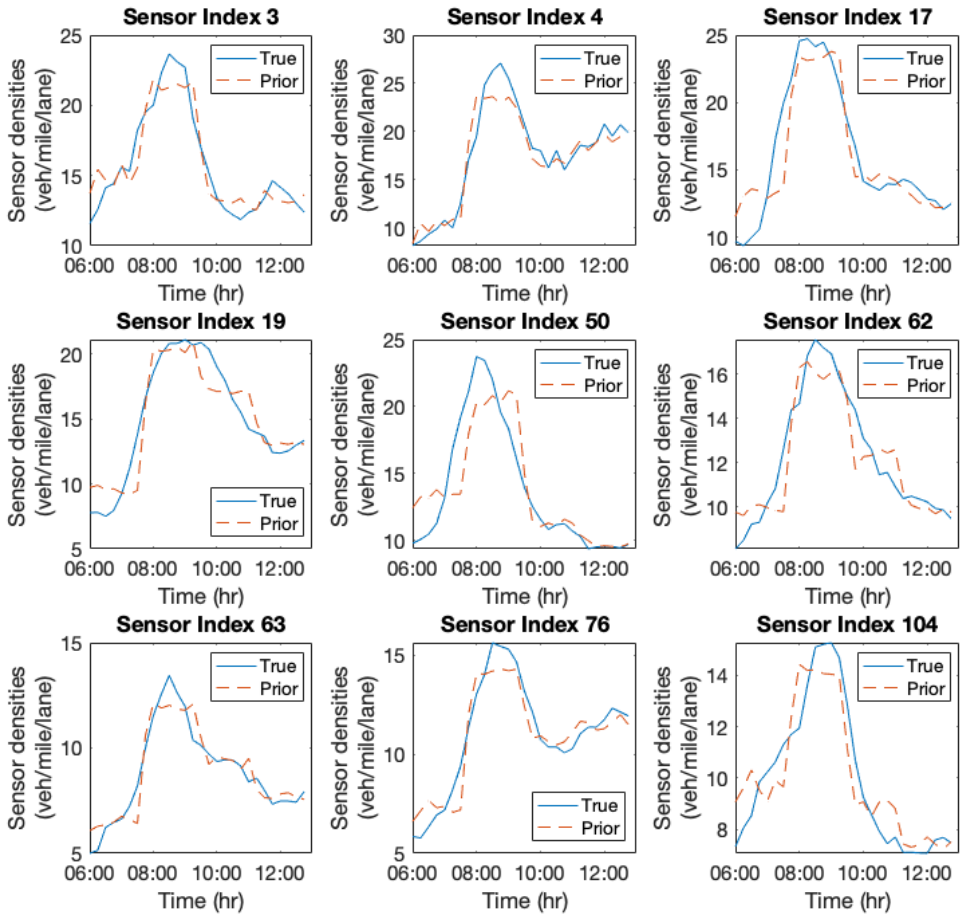

Figure 5-6: Comparison of sensor densities for true and prior demand profiles for subset of sensors in Singapore case study

given by the blue solid line, while the observed measurements for the prior demand are given by the red dashed line. Figure 5-6 provides the same plots but for density measurements (in vehicles per mile per lane) for the same subset of link sensors. While the plotted densities provide a useful snapshot of the demand scenarios, note that sensor densities are not used as field measurements in the online calibration process of this case study. As the figures show, both sensor counts and densities for the two demand profiles show similar trends to the trends of the O-D demand. More specifically, "observed" counts for both true and prior demand scenarios increase until peak period then start to dissipate.

Figure 5-5 also indicates that the sensor counts observed for the prior demand scenario already provide a good fit to the observed sensor counts for the true demand scenario, though they underestimate the exact peaks of the true counts. Still, this indicates that the baseline performance of the "offline-calibrated" prior demand may be difficult to improve upon for this case study. An online calibration algorithm given field measurements not radically different from those corresponding to the prior demand will not need to make very large adjustments to the parameter estimates. Figure 5-6 shows a similar situation in terms of link densities. The link densities at the same subset of sensors range from close to zero to a maximum of approximately 28 vehicles per mile per lane. These observed densities indicate very little to zero congestion on the network. In addition, the difference between the densities for the true and prior demand scenarios is small. It is interesting to note that even with a synthetically constructed prior demand that shows large deviations from the true demand, the sensor counts and densities on the Singapore network remain similar—the difficulties of creating a large-scale case study using synthetic demand and its implications on performance evaluation are discussed in more detail in Section 6.2.2.

5.1.3 Experimental design

Many of the same experimental design specifications from the Florian network case study detailed in Section 4.1.4 directly apply to the Singapore expressway network case study as well. One major difference is a one-hour warm-up period (from 5:00am to

6:00am) is used to ensure that the network is adequately loaded before the simulation period for online calibration begins at 6:00am. Once the warm-up period ends, all algorithms start with an initialization period of 15 minutes where O-D parameters are estimated using the generalized least-squares (GLS) approach. The Kalman filtering approaches are initialized in the same manner as for the Florian case study. The same state-space model formulation given in Equation 4.3 is used, as are the same EKF algorithm parameters ($a = 0.4$, $q_0 = 1$, $b_1 = 0.03$, $r_{0,1} = 1$, $b_2 = 0.6$, $r_{0,2} = 1$, and $w = 100$).

For the Metamodel EKF approach, the analytical network model detailed in Section 3.3 and a linear polynomial correction component are used in the formulation. In order to limit the number of metamodel parameters that need to be estimated and to avoid overfitting due to overparameterization, only O-D demand parameters that have a possible route through the sensor corresponding to measurement j are included in the linear polynomial component of the metamodel formulation corresponding to observed measurement j. Unless specified (Section 5.2.5), no simulator-evaluated points are generated during the offline phase of the MM-EKF algorithm to fit the metamodel.

5.1.4 Performance benchmarks

The Metamodel EKF algorithm (MM-EKF) is compared to the same benchmarks (GLS, Assign-EKF, FD-EKF) as in the Florian network case study, detailed in Section 4.1.5. Since the Singapore network case study calibrates O-D demand parameters given sensor count field measurements, the GLS and Assign-EKF benchmarks can be applied. Still, as addressed in the previous chapter in Section 4.3, both of these methods are specialized (and thus perform well) for this particular online calibration problem and are not generalizable to other online calibration problems with different calibration parameters or field measurements in the same way the Metamodel EKF algorithm is. This aspect of the comparison with benchmark algorithms is not addressed in this chapter.

One additional benchmark algorithm is shown for this case study. As discussed in

Section 1.3, the Limiting EKF (LimEKF) algorithm, first proposed by Antoniou [3], is derived from the FD-EKF benchmark algorithm and is motivated by the difficulty in scaling the FD-EKF algorithm for use on a high-dimensional problem due to the computational cost of the central finite differences numerical gradient approximation method. The Limiting EKF algorithm calculates an average gradient matrix offline from previously estimated numerical gradients and applies the same average gradient matrix at each time step of the EKF calibration. For the Singapore network case study, the average gradient matrix is computed from the gradient matrix estimates produced online during the FD-EKF calibration run. In comparing the proposed Metamodel EKF algorithm to both the FD-EKF and LimEKF algorithms, the difficulties in scaling an extended Kalman filter approach for use on a large-scale network are illustrated. The two benchmarks represent opposite poles on the spectrum of numerical gradient estimation techniques. The FD-EKF algorithm has a high online computational cost and can be difficult to tune with respect to simulator stochasticity, but theoretically provides the best localized gradient approximation. On the other hand, the LimEKF algorithm shifts the computationally expensive gradient approximation step offline with a trade-off of worse online performance in calibration accuracy. Note that the proposed MM-EKF algorithm, along with the other Kalman filter algorithms (Assign-EKF, FD-EKF, LimEKF), are identical in implementation except for how they each approximate the gradient in the linearization step of the Kalman filter. Comparisons among these online calibration algorithms will illuminate the pros and cons of each gradient approximation technique.

For the Singapore network case study, the only field data used to perform the online calibration is sensor count data. Calibration performance is measured using the same experimental design as the Florian network case study, specified in Section 4.1.4, though the FD-EKF and LimEKF benchmarks are each only run once, due to time limitations. The performance of each algorithm is assessed using the root mean-square normalized (RMSN) error given in Equation 4.5.

5.1.5 Summary

To summarize, the online calibration experiments for the Singapore network case study have the following attributes:

- Simulation period from 6:00am to 1:00pm with 15-minute calibration intervals and an hourlong warm-up from 5:00am to 6:00am,
- State vector of calibration parameters is of dimension 4050 consisting of O-D demand parameters,
- Measurement vector is of dimension 172 consisting of sensor counts,
- Algorithms run are the proposed MM-EKF and benchmarks GLS, Assign-EKF, FD-EKF, and LimEKF,
- Estimated and predicted (both 1 time interval into the future and 2 time intervals into the future) sensor counts and densities are generated for all online calibration results,
- Performance is evaluated using the root mean-square normalized (RMSN) error metric.

5.2 Results

This section presents the results of the case study. Validation of the analytical model on the Singapore network is presented in Section 5.2.1 to illustrate the modeling power of the network-specific traffic model for O-D demand detailed in Section 3.3 on a large-scale network. Summary results for online calibration are then shown in Section 5.2.2 to demonstrate the overall performance of the proposed MM-EKF approach in estimation and prediction compared to benchmarks. A deeper dive into the algorithm's performance is then provided, with results broken down by hour in Section 5.2.3, and computational results are shown in Section 5.2.4. Some initial

empirical results for different methods of fitting the metamodel are provided in Section 5.2.5 before the chapter concludes with final results for the Metamodel EKF algorithm in Section 5.2.6 and some discussion of the challenges of applying the algorithm to a large-scale problem.

5.2.1 Validation of analytical model

The analytical traffic model presented in Section 3.3 is validated on the Singapore expressway network in order to evaluate how it performs on a large-scale network. While there are 924 nodes and 1150 links in the simulator model of the Singapore expressway network, the analytical traffic model for O-D demand on the network only models a subset of them (265 nodes and 860 links). The link subset is chosen based on the links comprising a pre-determined route choice set used by the analytical model; each of the 172 sensored links falls within the subset of links modeled by the analytical model. The fixed route choice set was created from DynaMIT parameter files of the Singapore network representation. This model is validated on the Singapore network using the time-dependent O-D demand scenario presented in Section 5.1.2. For each 15-minute time interval in the simulation period from 6:00am to 1:00pm, the analytical traffic model is solved with the O-D demand as input, and both link count and density outputs are recorded.

Figures 5-7 through 5-10 show the comparison of the analytical traffic model outputs with the "true" observations for the synthetic demand scenario output by the simulator. Figure 5-7 shows the fit of analytical counts on the y-axis (in vehicles) to the simulator output on the x-axis (in vehicles). Each sensor in the network is represented by a different color; the blue line gives the 45-degree line that represents perfect correlation. The correlation coefficient for sensor counts is 0.98, which indicates a strong linear relationship between the two datasets. The scatterplot also confirms this; most points are clustered tightly around the 45-degree line while a few sensors seem to be mis-estimated by a constant factor, although the relationship is still linear. The analytical traffic model is able to replicate the transient simulator counts very closely although it does not model transient dynamics itself. Figure 5-8

confirms this analysis in greater detail for the same subset of link sensors as in Figure 5-5. Each plot graphs sensor counts on the y-axis output by both the simulator and analytical model for the simulation period represented on the x-axis. The blue solid curve gives simulator-generated counts, and the red dashed curve graphs analytical counts. For the majority of the sensors shown in Figure 5-8, the two curves are near-identical. For sensor 104, the analytical traffic model is able to reproduce the overall shape of the simulator counts but systematically overestimates the magnitude; the scaling factor difference can be corrected in the metamodel formulation with the parameter α_j^h in Equation 3.1. From this comparison of simulator and analytical model sensor counts, it can be expected that the metamodel approximation, if not the analytical traffic model by itself, can successfully reproduce simulator outputs for the Singapore network.

Similar plots for sensor densities are given in Figures 5-9 and 5-10. Figure 5-9 shows several sensors with density observation points clustered around the blue 45-degree line, though there are several other sensors which diverge from this trend. For any specific sensor (color of point), the relationship between the simulator and analytical model densities looks to be strongly linear, even when not in line with the 45-degree line and instead off by a constant scaling factor. The overall correlation coefficient for the density measurement datasets is 0.79, which implies less of an overall linear relationship than for count measurements. In general, the biggest discrepancies between the analytical model and the simulator are caused by cases where link densities are overestimated by the analytical model. As seen in the subset of sensors plotted in Figure 5-10, there are a few sensors that largely get the trend and magnitude correct (sensor indices 17, 63, 76). For the other sensors, the shape of the analytical model curve mirrors that of the simulator curve; the affine scaling discrepancy can be corrected through the fitting of the metamodel parameters. For a more systematic correction, the bias of the analytical model can also be corrected by fine-tuning the network parameters (i.e., jam density) of the analytical traffic model as formulated in Equations 3.5 through 3.10. While the case study in this chapter does not use sensor densities as field measurements, in general a better calibration

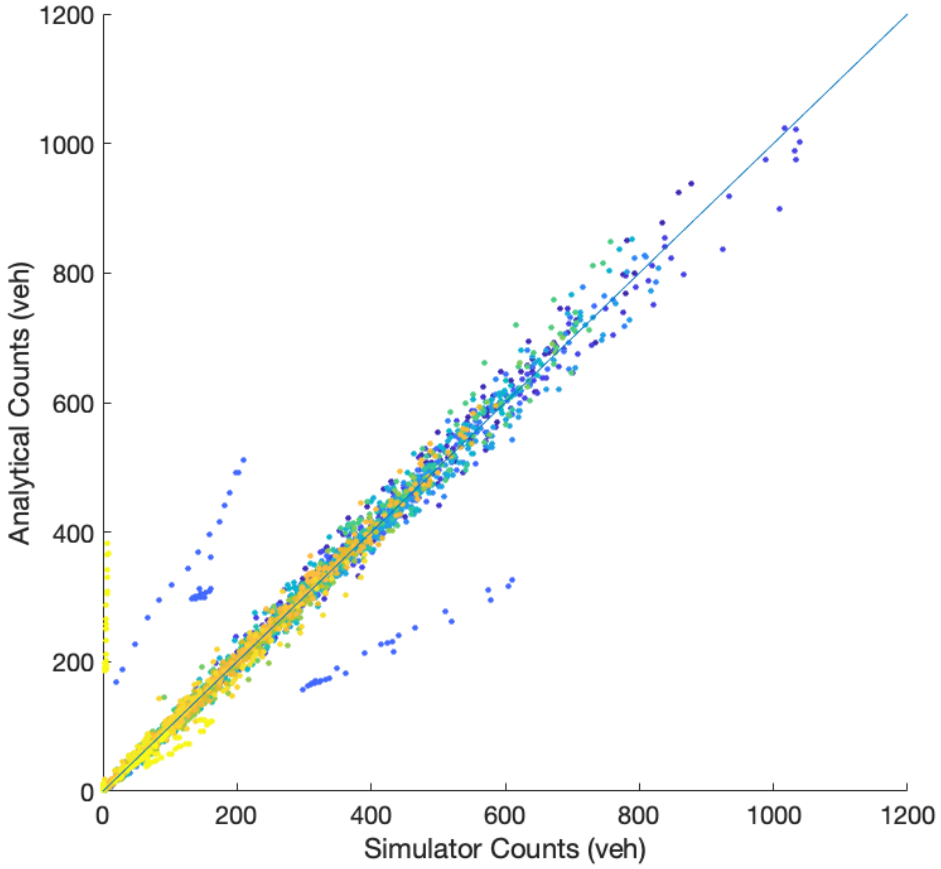

Figure 5-7: Scatterplot of simulated vs. analytical model sensor counts for Singapore case study

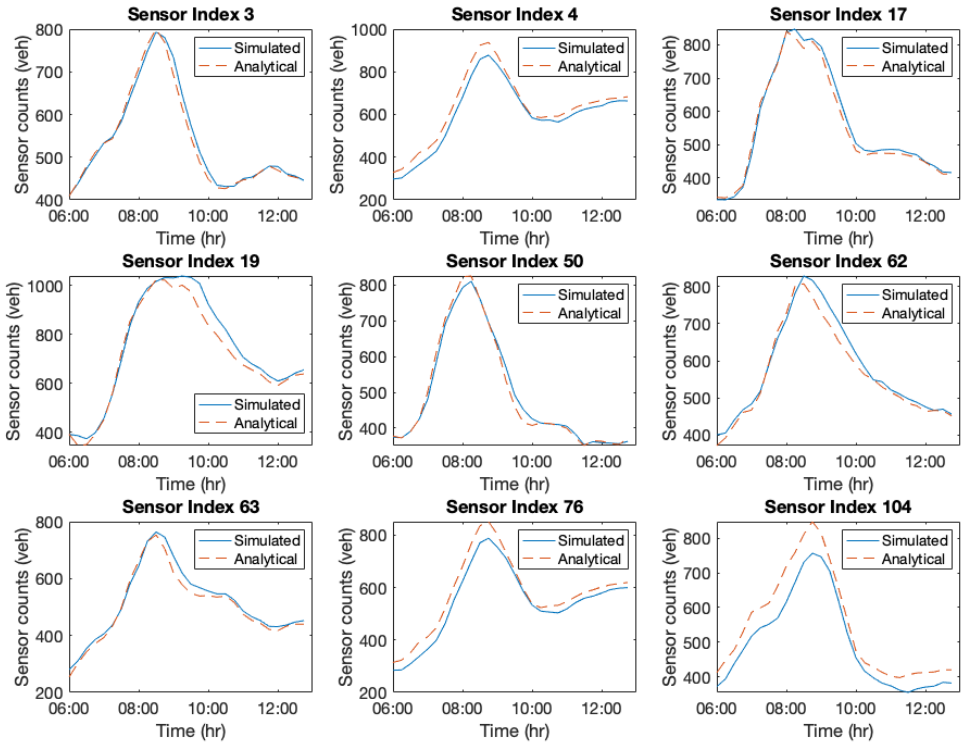

Figure 5-8: Comparison of observed counts for true and prior demand profiles on a subset of Singapore sensors

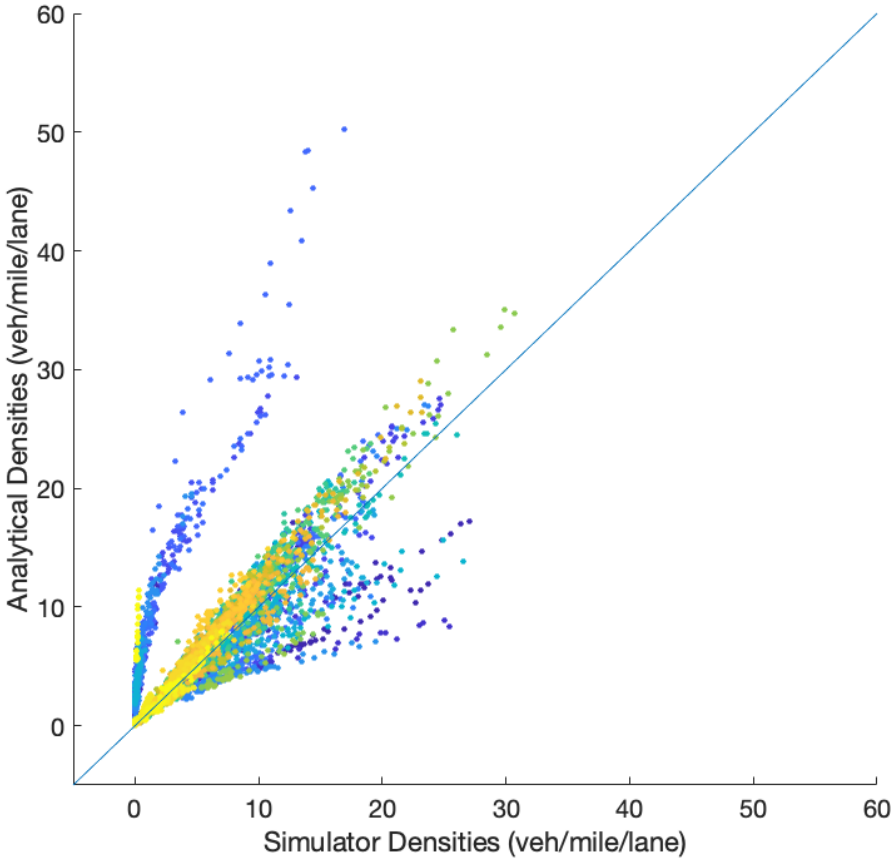

Figure 5-9: Scatterplot of simulated vs. analytical model sensor densities for Singapore case study

of the analytical model to the network may lead to less bias in the Metamodel EKF approach. Overall, the analytical traffic model does an impressive job of approximating traffic measurements output by the simulator on a large-scale network, even with such a high-level macroscopic representation of traffic behavior on the network.

5.2.2 Summary across simulation period

Since the primary stated objective of online calibration is accuracy of fit to field measurements, the algorithm performance for estimation of sensor counts is analyzed first. Tables 5.1 and 5.2 shows a summary of performance metrics for the estimation and prediction of sensor counts across all online calibration algorithms tested. The

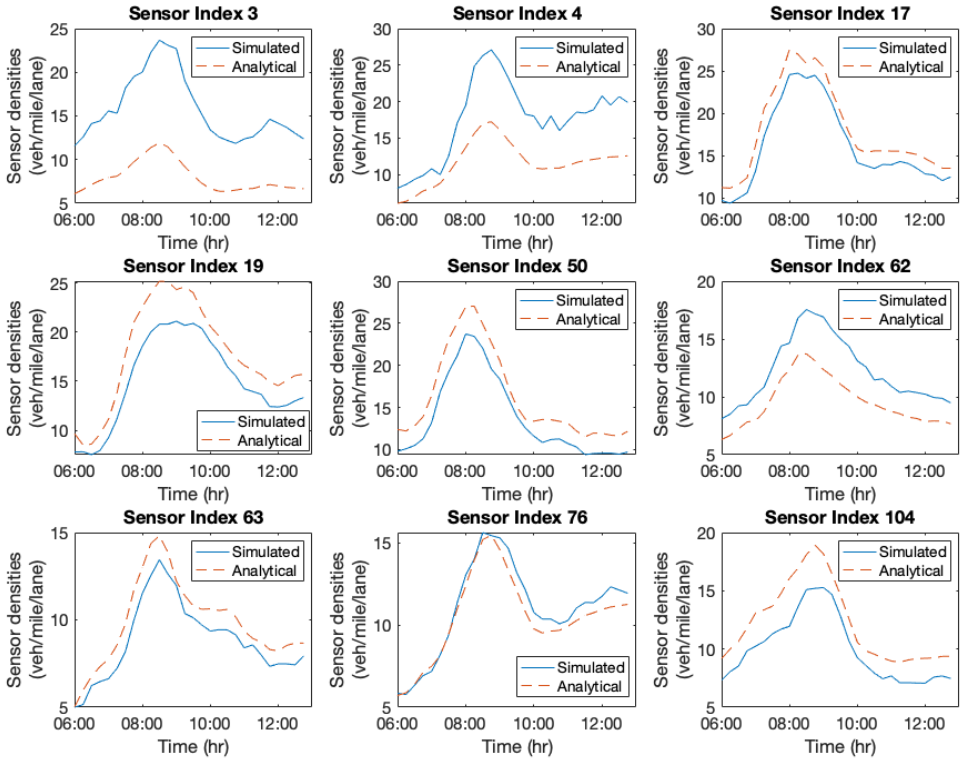

Figure 5-10: Comparison of observed densities for true and prior demand profiles on a subset of Singapore sensors

		Estimation		1-step prediction		2-step prediction	
Algorithm	Run	RMSN	% improv	RMSN	% improv	RMSN	% improv
Baseline		0.143	-	0.150	-	0.184	-
MM-EKF	1	0.082	42.7	0.152	-1.2	0.219	-19.1
	2	0.086	40.1	0.153	-2.0	0.224	-21.4
	3	0.085	41.0	0.150	0.1	0.220	-19.5
Assign-EKF	1	0.068	52.6	0.139	7.2	0.222	-20.8
	2	0.069	51.7	0.137	8.6	0.225	-22.1
	3	0.070	50.9	0.142	5.8	0.224	-21.4
GLS	1	0.039	73.0	0.149	1.1	0.214	-16.1
	2	0.037	74.0	0.143	5.1	0.170	7.6
	3	0.037	74.4	0.143	4.8	0.168	8.8
FD-EKF	1	0.288	-100.8	0.317	-111.1	0.345	-87.3
LimEKF	1	0.304	-111.9	0.336	-123.8	0.367	-99.5

Table 5.1: Performance metrics for estimation and prediction of sensor counts for online calibration algorithms in Singapore network case study

	Estimation		1-step prediction		2-step prediction	
Algorithm	RMSN	% improv	RMSN	% improv	RMSN	% improv
Baseline	0.143	-	0.150	-	0.184	-
MM-EKF	0.084	41.3	0.152	-1.0	0.221	-20.0
Assign-EKF	0.069	51.7	0.139	7.2	0.224	-21.4
GLS	0.038	73.8	0.145	3.7	0.184	0.1
FD-EKF	0.288	-100.8	0.317	-111.1	0.345	-87.3
LimEKF	0.304	-111.9	0.336	-123.8	0.367	-99.5

Table 5.2: Summary of estimation and prediction metrics for sensor counts in Singapore network case study

layout is similar to that of Tables 4.2 and 4.4 but focused specifically on sensor count fit. Each row of the table reports results for one calibration run of an algorithm; results for three runs of the MM-EKF, Assign-EKF, and GLS algorithms are reported, as well as one run of both FD-EKF and LimEKF algorithms. Table 5.2 shows the average performance metrics across all three algorithm runs of a particular algorithm. The RMSN values and percent improvement (calculated off of the "Baseline" row) are reported for fit to sensor counts in the estimation, 1-step prediction, and 2-step prediction stages.

First note that the performance of using the baseline of prior O-D demand parameters to fit sensor counts is already quite good, with a RMSN error of 14.3% for estimation. This was suggested in the discussion of Figure 5-5, as sensor counts for the true and prior demand scenarios were shown to be close for a subset of the link sensors. Given the good performance of the baseline to start, the online calibration algorithms may be motivated to only make small adjustments based on the observed sensor counts and estimate O-D parameters near the prior values. Still, several of the algorithms show an improvement over the baseline in estimating sensor counts. The average RMSN error for the MM-EKF algorithm in fitting observed counts is 8.4%, compared to average RMSN values of 6.9% for Assign-EKF and 3.8% for GLS; the corresponding percent improvements are 41.3% for MM-EKF, 51.7% for Assign-EKF, and 73.8% for GLS. To gauge the statistical significance of these improvements, a 95% confidence interval for each algorithm is calculated for the three-run sample.

The confidence interval for the MM-EKF error metrics is $(7.9\%, 8.9\%)$, which implies the improvement shown by the proposed algorithm over the prior O-D parameters in estimating sensor counts is statistically significant. One-sided Mann-Whitney tests also confirm that for estimation, the GLS algorithm performs the best followed by the Assign-EKF algorithm and then the MM-EKF algorithm. In terms of absolute RMSN error, all three of these algorithms show good performance with averages less than 10%, a threshold indicating a close fit as discussed in Section 4.1.5.

This corroborates what was seen in the Florian case study. In terms of estimating observed measurements from the field, the GLS algorithm produces smaller error metrics than the Kalman filter algorithms for the O-D calibration problem. This is not entirely surprising—for the Singapore network, the optimization problem involves estimating 4050 O-D parameters given 172 sensor count measurements, which is a very under-determined problem. The generalized least-squares algorithm is able to identify parameter values that minimize this objective with an average error of 3.8%. On the other hand, the Assign-EKF and MM-EKF algorithms impose structural restrictions on the optimization problem that the GLS algorithm does not through the state-space model formulation and specifically the transition equation. While both EKF algorithms still produce good solutions in fit to measurements, the parameters are estimated based on the estimates and covariance matrix of the parameters from the previous time step in addition to sensor count measurements from the current time step. As a result, the GLS algorithm has more of a tendency to overfit to measurements than the Kalman filtering approaches, which leads to good performance in estimation but worse performance in prediction. This is confirmed in the discussion below.

More interestingly, the Assign-EKF algorithm shows better performance than the proposed MM-EKF algorithm in estimation of sensor counts. The sole difference in the algorithms is in the gradient estimation method, though bias in the case study experimental design mentioned in Section 4.1.4 may also help explain this trend. The bias is introduced in the tuning of the Kalman filter parameters. The Kalman filter parameters for the case studies were tuned prior to the online calibration experiments

using the Assign-EKF algorithm, primarily due to the faster runtime of the algorithm. Thus the grid search used to tune the Kalman filter parameters found the optimal setting for the Assign-EKF algorithm, which is not necessarily the optimal parameter setting for the MM-EKF (or FD-EKF or LimEKF, for that matter) algorithm. Ideally, the Kalman filter parameters would be tuned for each individual algorithm, though other considerations like overfitting and data availability should also be taken into account, some of which are addressed in Section 6.2.2. Still, the divergence in performance between the Assign-EKF and MM-EKF approaches points to differences in the gradient estimation methods. In particular, it seems the metamodel approach for approximating the gradient does not perform as well for the large-scale Singapore network. Contrasting this with the strong performance of the analytical traffic model in fitting sensor counts in Section 5.2.1, it may be that the metamodel can replicate simulator output well but not the gradient. Since the metamodel is fit using only simulator output and no first-order information from the simulator, there is no explicit guarantee that the analytical gradient of the metamodel is an accurate approximation of the simulator gradient. As the extended Kalman filter approach needs a good estimate of the gradient at each time step to perform optimally, further exploration of the attributes of the metamodel gradient approximation and methods for improving the first-order fit of the metamodel should be considered in future research.

On the other hand, the performance metrics for the FD-EKF and LimEKF algorithms in estimating sensor count measurements are worse than the baseline of prior O-D demand. The RMSN error for the FD-EKF algorithm run is 28.8%, while the RMSN error for the LimEKF algorithm run is 30.4%. As expected from the discussion in Section 5.1.4, the LimEKF algorithm shows a slightly worse performance in calibration accuracy compared to the FD-EKF but the online computational cost is significantly reduced (see Section 5.2.4). As for the poor performance of these two algorithms in comparison to the baseline and other online calibration algorithms, there are a few possible reasons for this deterioration in performance. First, there was difficulty in fine-tuning the FD-EKF algorithm for the large-scale Singapore network. Whereas the MM-EKF and Assign-EKF methods for estimating the gradient were

straightforward to apply and achieved good performance without much testing of the gradient calculation procedure, the algorithm parameters for the finite differences gradient estimation used by FD-EKF were more difficult to calibrate. Coupled with the high computational cost of running even one pass of the FD-EKF algorithm, implementation of the algorithm for a network of this size proved much more unwieldy than comparable EKF algorithms.

The other element that may factor into the poor performance of the FD-EKF algorithm is the impact of simulator stochasticity on the gradient approximation method. This has been detailed already in Section 4.3.4, but the effect is felt more strongly in the Singapore network case study as only one replication of the gradient estimation step is taken due to the computational limitations of the problem. While the FD-EKF does produce an unbiased estimate of the gradient where the MM-EKF does not, it is also an especially noisy estimate due to its reliance on thousands of evaluations of the stochastic traffic simulator. As a result, the FD-EKF algorithm, as well as the LimEKF algorithm, which relies on the gradient estimations from FD-EKF, suffer in performance accuracy. Ultimately, for the Kalman filter parameter settings that were used for this case study, the Assign-EKF and MM-EKF algorithms were able to achieve an improvement in performance compared to the baseline as measured by the RMSN error metric in estimating sensor counts. The FD-EKF and LimEKF benchmarks were not able to achieve similar results. Better performance for the FD-EKF and LimEKF algorithms may be possible with further adjustments, though compared to other Kalman filter approaches like the Assign-EKF and MM-EKF algorithms, they seem to require more effort to achieve favorable results for a large-scale application. Given the difficulty of implementing the finite differences approach for gradient estimation, it would also be interesting to apply other numerical gradient estimation methods (i.e., simultaneous perturbation) in future case studies of this scale as a benchmark for the proposed MM-EKF approach.

Tables 5.1 and 5.2 also show experiment results for the 1-step and 2-step prediction stages. Since the online calibration algorithms are run for 15-minute time intervals in this case study, one-step prediction statistics correspond to prediction for 15 minutes

into the future, while two-step prediction statistics correspond to prediction for 30 minutes into the future. Positive values in the percent improvement columns indicate performance better than the baseline, while negative values indicate performance worse than the baseline.

Overall, the online calibration algorithms show little to no improvement over the baseline. The proposed MM-EKF algorithm reports an average error metric of 15.2% for one-step prediction, marginally (1%) worse than the baseline, and 22.1% for two-step prediction, 20% worse than the baseline. The corresponding 95% confidence intervals are calculated to be $(14.8\%, 15.6\%)$ and $(21.5\%, 22.7\%)$, respectively, which indicate the performance of the MM-EKF algorithm is statistically indistinguishable from the baseline for 1-step prediction and significantly worse for 2-step prediction. The Assign-EKF and GLS benchmarks outperform the MM-EKF in the 1-step prediction stage, with statistical tests confirming the relative best performance of Assign-EKF (average RMSN error of 13.9%), followed by GLS (14.5%). Though the GLS algorithm shows a 3.7% improvement over the baseline, the 95% confidence interval is calculated to be $(13.6\%, 15.3\%)$, which indicates the difference is not statistically significant. In predicting further into the future (2-step prediction), the two primary EKF algorithms (MM-EKF and Assign-EKF) show considerable deterioration in performance compared to the baseline. The GLS algorithm (RMSN error of 18.4%) shows comparable results to the baseline. On the other hand, the FD-EKF and LimEKF benchmark algorithms continue to show poor performance in the prediction stages, with RMSN values of 31.7% and 33.6% for 1-step prediction, respectively, with worse performance for 2-step prediction.

There are a couple of takeaways to note from the prediction results. First, while the GLS benchmark performs better than the MM-EKF algorithm for prediction of sensor counts on the Singapore network, the magnitude of the divergence in performance shrinks considerably between estimation metrics and 1-step prediction metrics. This speaks to the over-performance of GLS in estimation compared to prediction due to overfitting to field measurements. Whereas the GLS benchmark shows a 73.8% improvement on the baseline for estimation, it only shows a (not statistically sig-

nificant) 3.7% improvement for 1-step prediction. Second, as mentioned previously, some of the better performance of Assign-EKF compared to MM-EKF may be due to experimental design bias. In looking at the 2-step prediction results for Assign-EKF and MM-EKF, where the Kalman filter parameter tuning bias has less effect, both algorithms perform comparatively. The poor performance of these Kalman filter approaches in prediction is discussed more below, and also in Section 6.2.1. Finally, the algorithm parameters were tuned according to a grid search based on estimation performance, not predictive performance. The degradation of the performance of the online calibration algorithms in the prediction stages signals that the algorithm parameters may be overfit for performance in estimation of sensor counts and should instead be better tuned for predictive performance.

For this set of prediction metrics, the baseline of prior O-D demand parameters also proves to be a difficult one to improve upon. The RMSN values for 1-step and 2-step prediction of sensor counts for the baseline are 15.0% and 18.4%, respectively. These are already good results for prediction; similar case studies found in the literature report prediction errors for historical parameters in the range of 36% to 42% [52, 73]. Part of the reason the baseline performs well is due to the specific nature of how the prior demand scenario was constructed—since the prior demand values are kept constant across four broad time windows that partition the simulation period, predictions using the prior demand are very similar to estimations (as confirmed by the RMSN values), and there is less deterioration in performance moving from the estimation stage to the prediction stages for the baseline. The impact of the prior demand on predictive performance was also seen in other experiments done for this case study, which are not reported. Using a different prior demand scenario for the same set of true O-D parameter values led to improved predictive performance relative to the baseline of prior demand. While these results are largely exploratory, the effect of prior information on algorithm performance is worth further analysis.

Most significantly, it is crucial to note that while predictive performance of the MM-EKF algorithm is an important objective in order to provide accurate prediction and guidance capabilities in a real-time setting (which is why it is reported), the

proposed algorithm makes no claim to improve forecasting power compared to other online calibration Kalman filtering algorithms. The metamodel approximation, the main development of the MM-EKF algorithm, is expressly used to embed structural information and reduce the computational burden of the gradient approximation in the EKF approach. While better predictive performance would be a welcome auxiliary effect of using the metamodel to estimate the underlying parameters, most of the predictive performance of EKF methods can be attributed to the assumptions made in the state-space model formulation. Prediction for a Kalman filtering approach is carried out by running the state estimates (i.e., O-D demand parameters) through the transition equation, which in this case is a simple random walk. For the results of this case study reported in Tables 5.1 and 5.2, the deterioration of both the Assign-EKF and MM-EKF algorithms in average RMSN error values shifting from estimation to prediction (and particularly 2-step prediction) confirms the influence of the state-space model on performance.

While broader state-space model considerations are addressed in Section 6.2.1, a simple random walk transition equation has major limitations for the Singapore network case study. As noted by Zhou and Mahmassani [75], the simple random walk can be effective for processes that change slowly but may not be rich enough of a model to capture the non-linear trends in dynamic O-D flow. In addition, for a large-scale network where trips take significantly longer than the time step of the online calibration problem, vehicles represented in the O-D demand parameters of one time interval have an impact on link measurements for several subsequent time steps. A simple random walk transition equation is unable to model this behavior in the Singapore network. This is in contrast to the prediction results for the Florian network in the previous chapter. In the Florian toy network, trip travel times fell within 15-minute calibration time interval, so a simple random walk transition equation was an adequate model and able to produce good results for predictive performance. The same is not seen to be true in the Singapore network case study, and the failure of the model to represent those dynamics leads to worse performance in the empirical results. As discussed in the Section 1.3, several methods for dealing with this complication

		densities						O-D	
		Estimation		1-step prediction		2-step prediction		Estimation	
Algorithm	Run	RMSN	% improv	RMSN	% improv	RMSN	% improv	RMSN	% improv
Baseline		0.364	-	0.376	-	0.397	-	1.172	-
MM-EKF	1	0.329	9.8	0.368	2.1	0.466	-17.4	1.387	-18.4
	2	0.322	11.7	0.384	-2.2	0.462	-16.3	1.518	-29.6
	3	0.335	8.0	0.370	1.5	0.435	-9.7	1.532	-30.7
Assign-EKF	1	0.329	9.8	0.372	1.0	0.447	-12.5	1.150	1.8
	2	0.322	11.7	0.368	2.1	0.436	-9.9	1.155	1.4
	3	0.327	10.3	0.362	3.8	0.425	-7.2	1.154	1.5
GLS	1	0.327	10.4	0.380	-1.1	0.454	-14.3	1.762	-50.4
	2	0.324	11.0	0.385	-2.5	0.416	-4.9	1.106	5.6
	3	0.320	12.2	0.394	-4.8	0.423	-6.6	1.105	5.7
FD-EKF	1	0.453	-24.3	0.496	-32.0	0.530	-33.5	2.161	-84.4
LimEKF	1	0.475	-30.3	0.507	-35.0	0.561	-41.4	2.094	-78.7

Table 5.3: Performance metrics for estimation and prediction of sensor densities and O-D parameters in Singapore network case study

	densities						O-D	
	Estimation		1-step prediction		2-step prediction		Estimation	
Algorithm	RMSN	% improv	RMSN	% improv	RMSN	% improv	RMSN	% improv
Baseline	0.364	-	0.376	-	0.397	-	1.172	-
MM-EKF	0.329	9.9	0.374	0.5	0.454	-14.5	1.479	-26.3
Assign-EKF	0.326	10.6	0.367	2.3	0.436	-9.9	1.153	1.6
GLS	0.324	11.2	0.386	-2.8	0.431	-8.6	1.324	-13.0
FD-EKF	0.453	-24.3	0.496	-32.0	0.530	-33.5	2.161	-84.4
LimEKF	0.475	-30.3	0.507	-35.0	0.561	-41.4	2.094	-78.7

Table 5.4: Summary of estimation and prediction metrics for sensor densities and O-D parameters in Singapore network case study

have been proposed, including using a transition equation with a higher autoregressive degree in the state-space model formulation and expanding the state to include O-D demand parameters for both the current time interval and several previous time intervals. Both of these methodologies increase the online computational burden of the solution algorithm, but for a large-scale network like the Singapore network, the trade-off should be considered and may be worth it for better predictive performance.

Tables 5.3 and 5.4 report experiment results for both link density estimation and prediction and O-D parameter estimation. For density estimation metrics, the MM-EKF algorithm produces similar results to the Assign-EKF and GLS benchmarks, and all three algorithms show a small improvement over the baseline. The corresponding average RMSN values for estimated density measurements are 32.9% for MM-EKF, 32.6% for Assign-EKF, and 32.4% for GLS. In addition, Mann-Whitney tests show

no statistically significant difference among the performance of the three algorithms, though all three show significant improvement over the baseline. Again, the FD-EKF and LimEKF algorithms show poor performance compared to the baseline. Similarly to with the sensor count metrics, all online calibration algorithms deteriorate in performance moving from estimation to prediction. While some of this is due to a good baseline to start, the factors mentioned previously that impact predictive performance also hold in this case. In addition, the algorithms do not explicitly calibrate for link density measurements, so fit to densities in estimation is an auxiliary metric for performance.

The performance metrics in the rightmost columns of Tables 5.3 and 5.4 for fit to the underlying O-D parameters show that the MM-EKF algorithm does not improve on the baseline of prior O-D parameters. The average RMSN error for MM-EKF is 147.9%, compared to 117.2% for the baseline. Comparatively, the average RMSN error for the benchmarks algorithms are 115.3% for Assign-EKF (a slight improvement over the baseline), 132.4% for GLS, 216.1% for FD-EKF, and 209.4% for LimEKF. Interestingly, while both the MM-EKF and GLS algorithms significantly improve over the baseline in terms of fit to sensor count measurements, they do so by identifying O-D parameter values with larger error metrics. Since fit to the underlying true parameter values is not an explicit objective of the online calibration optimization, and for real-world problems the true parameter values would not be known anyway, this result is not necessarily a red flag. Regardless, it does point to poor predictive performance seen in these empirical results. As the dynamic O-D matrices estimated by these methods do not resemble the true O-D matrices, the state-space model does not model the temporal evolution of demand on the network accurately, which leads to worse performance in traffic state prediction.

Figure 5-11 summarizes the performance of the different algorithms for estimation, 1-step prediction, and 2-step prediction of sensor counts in a bar graph. The format is the same as Figure 4-11. Each bar represents the RMSN error averaged across the three runs of the algorithm (for the MM-EKF, Assign-EKF, and GLS algorithms). Error bars provide 95% confidence intervals calculated from the three algorithm runs;

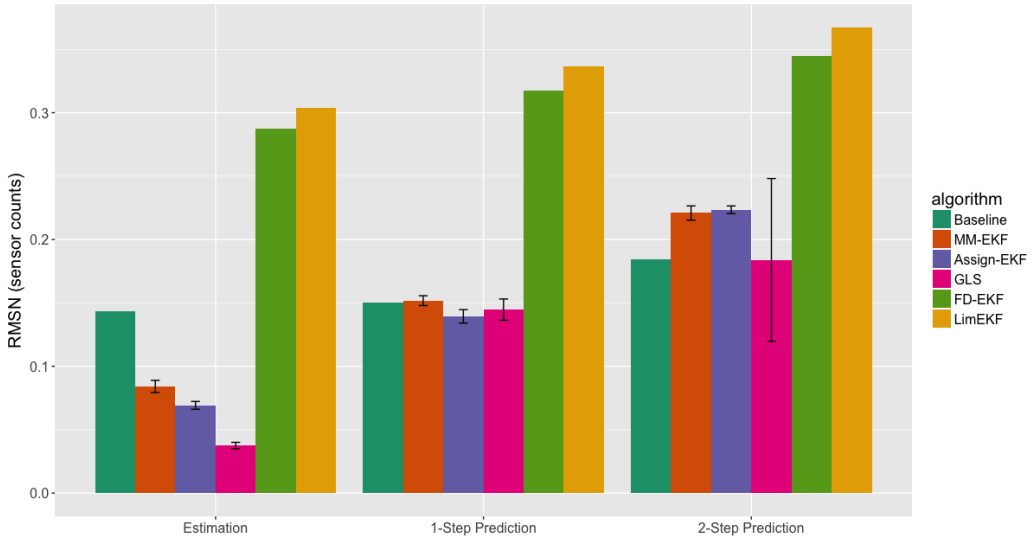

Figure 5-11: Average performance metrics for sensor counts for MM-EKF algorithm and benchmarks for Singapore case study

note that there are no confidence intervals for FD-EKF and LimEKF since each is only run once. The orange bar in each grouping represents the results of the proposed MM-EKF algorithm. The main takeaways from the above analysis are illustrated in Figure 5-11. For estimation of sensor counts, the GLS algorithm performs the best among all online calibration approaches, though the MM-EKF and Assign-EKF algorithms also improve on the baseline. For prediction, the baseline deteriorates the slowest as explained previously; among the online calibration algorithms GLS loses the comparative advantage it shows in estimation. The error metric for the GLS algorithm in 2-step prediction of sensor counts exhibits a wide confidence interval, indicating variability among the three algorithm runs in predictive performance. The deterioration in predictive performance is especially stark as shown in the figure, as well as the poor performance of the FD-EKF and LimEKF benchmark algorithms.

Figure 5-12 presents a similar bar graph for density metrics. For density error metrics, the bars are more closely clustered than the corresponding results for sensor counts, and differences between algorithms are much smaller. In terms of estimated densities, the MM-EKF algorithm outperforms the baseline, but as noted previously, its performance is statistically indistinguishable from the two benchmark algorithms,

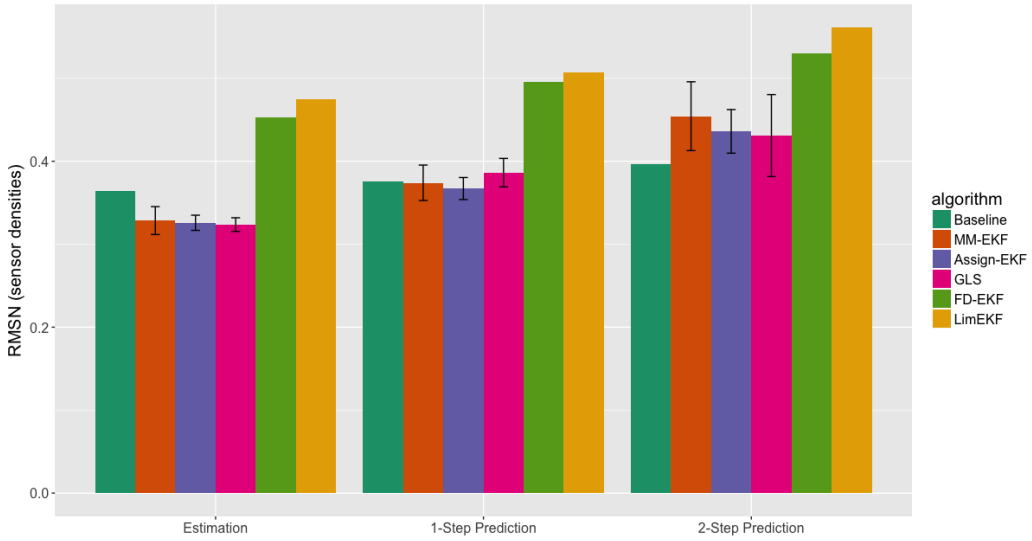

Figure 5-12: Average performance metrics for sensor densities for MM-EKF algorithm and benchmarks for Singapore case study

GLS and Assign-EKF. In general, the results for prediction of sensor densities show little to no improvement over the baseline, though the FD-EKF and LimEKF algorithms continue to perform poorly compared to the other online calibration algorithms. As seen in Figure 5-6, density measurements for the true and prior demand scenarios are generally very similar for the subset of sensors plotted, and given that the online calibration algorithms do not receive density measurements as field data, improvement on those prior density values is not expected.

5.2.3 Results by time interval

A closer look at the hour-by-hour breakdown of the performance in estimation and prediction of sensor counts provides a little more clarity on the accuracy of the proposed MM-EKF algorithm. Hourly RMSN metrics, calculated in the same manner as in Section 4.2.3, highlight the temporal aspects of each algorithm's performance. Figure 5-13 gives average RMSN metrics per hour for estimation of sensor counts for each of the online calibration algorithms. Each group of bars represents an hour of the simulation period as labeled on the x-axis, and each bar reports the average RMSN value for the algorithm. Error bars give the 95% confidence intervals calcu-

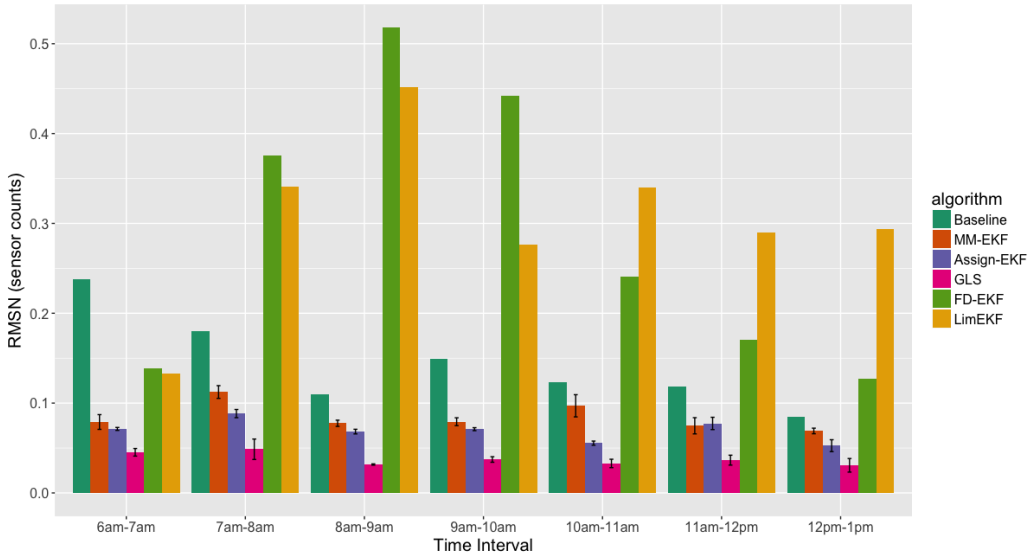

Figure 5-13: RMSN error metrics by hour for estimated sensor counts for Singapore case study

lated for each set of three algorithm runs for the MM-EKF, Assign-EKF, and GLS algorithms. As seen in the figure, the GLS algorithm is the best performer across every hour of the simulation period, followed by the MM-EKF and Assign-EKF algorithms. The proposed MM-EKF algorithm outperforms the baseline in every hour and shows consistent performance across the entire simulation period, as do the Assign-EKF and GLS algorithms. While the FD-EKF and LimEKF algorithms perform poorly across the simulation period, the FD-EKF algorithm shows more temporal variability. Errors for the FD-EKF algorithm peak during the hour from 8am to 9am, representing morning peak period, before decreasing to more reasonable RMSN values. The LimEKF, which also shows poor performance, uses a fixed average gradient estimate across all time intervals and thus shows more consistent RMSN values. Both FD-EKF and LimEKF algorithms show better performance than the baseline during the hour from 6am to 7am before divergence of the EKF algorithm due to poor gradient approximation leads to poor performance in estimation during the rest of the simulation period.

Figures 5-14 and 5-15 show similar metrics for the 1-step and 2-step prediction stages. The graph is set up similarly to Figure 5-13. Note that the first set of bars in

each figure are for the periods 6:15am to 7:00am and 6:30am to 7:00am, respectively, since 1-step and 2-step predictions are not available for the periods 6:00am to 6:15am and 6:00am to 6:30am, respectively. The general trends are similar for the two graphs. The better performance of the GLS algorithm in estimation has largely disappeared in the 1-step prediction results, and the algorithm performs similarly to the Assign-EKF algorithm. In most hours of the simulation period, the MM-EKF algorithm performs as well as the GLS and Assign-EKF benchmarks but in others, particularly from 10am to 11am, it does poorly even compared to the baseline. The baseline predictions alternate between low and high errors due to the manner in which the prior demand profile was constructed, with a substantial amount of variation across the simulation period. The predictive performance of the baseline is especially good from 12pm to 1pm, when the true demand profile has largely stabilized as seen in Figure 5-3, and the prior demand is very similar to the true demand. The 2-step prediction metrics make these takeaways even more apparent. The MM-EKF and Assign-EKF algorithms behave similarly, as performance in prediction is largely dependent on the shared state-space model formulation, and perform worse on average than GLS and often the baseline as well. Again, the MM-EKF algorithm has particular trouble with the hour from 10am to 11am, but otherwise performance is consistent across hours. Overall, the performance of the online calibration algorithms for prediction on the Singapore expressway network has major limitations, and the considerations outlined above should be addressed to improve the predictive performance of these methods.

5.2.4 Computational performance

A significant motivator for the development of the Metamodel EKF algorithm was to improve the computational efficiency of the black-box EKF framework as originally proposed by Antoniou [3] by avoiding the need for numerical gradient calculations during the linearization step of the algorithm. Computational performance is an important consideration for online calibration algorithms, especially in considering feasibility of the approach for high-dimensional problems. This section considers the computational performance of the proposed MM-EKF algorithm against the perfor-

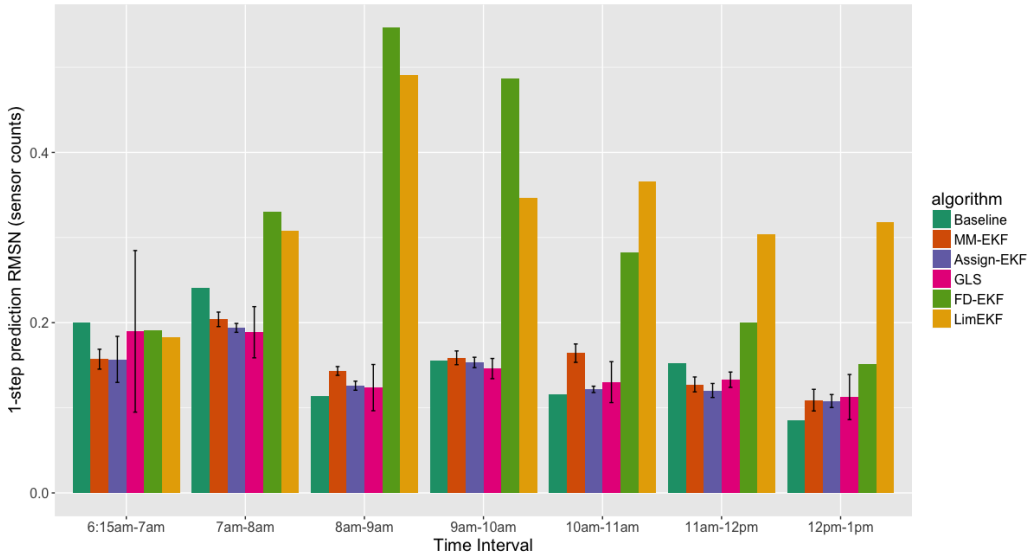

Figure 5-14: RMSN error metrics by hour for 1-step predicted sensor counts for Singapore case study

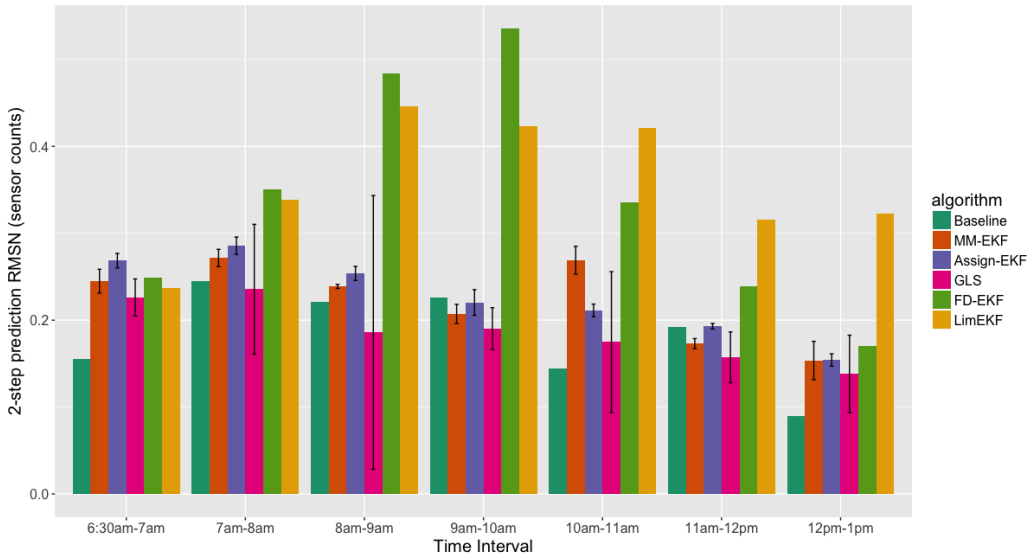

Figure 5-15: RMSN error metrics by hour for 2-step predicted sensor counts for Singapore case study

Algorithm	Run	Avg Runtime Per Interval (min)	Avg Simulator Eval Time (min)	Avg Gradient Calc Time Per Interval (min)
MM-EKF	1	4.3	3.3	0.39
	2	4.4	3.4	0.39
	3	4.3	3.3	0.38
	Avg	4.4	3.3	0.38
Assign-EKF	1	1.8	1.1	< 0.01
	2	1.7	1.1	0.01
	3	1.7	1.0	0.01
	Avg	1.8	1.1	0.01
GLS	1	6.5	N/A	N/A
	2	4.5	N/A	N/A
	3	5.9	N/A	N/A
	Avg	5.6	N/A	N/A
FD-EKF	1	921	1.2	918
LimEKF	1	2.1	1.2	0.01

Table 5.5: Summary of computational metrics for online calibration algorithms in Singapore network case study

mance of the benchmark algorithms for the large-scale Singapore network. In doing so, it validates that the MM-EKF approach can be used to calibrate parameters for a problem of dimension 4050 on a real-world network. As discussed in Section 1.3, this case study on the Singapore network ranks as one of the largest online calibration case studies seen in the literature reviewed. While analysis of computational performance for the Florian network in the previous chapter was largely a theoretical discussion given the fast runtime of all of the algorithms considered, analysis for the Singapore network has practical implications. Given the limited computational budget as defined by the 15-minute calibration intervals in this case study, the feasibility of the various algorithms is evaluated in terms of computational runtime in minutes.

Table 5.5 shows runtime statistics for the different online calibration algorithms run in this case study. Each row of the table reports runtime statistics for one algorithm run; the rows labeled "Avg" report the average results across the three algorithm runs for each of MM-EKF, Assign-EKF, and GLS. The results for FD-EKF and LimEKF are reported for the only run done for the case study. The first column of metrics measures the average total runtime (in minutes) of the online calibration

algorithm to identify parameter estimates for the 15-minute time interval, where the average is taken over the entire simulation period. The second column measures the average runtime (in minutes) for a single simulator evaluation (the time it takes to run the simulator for a set of O-D parameter values for a 15-minute time interval and record simulator measurements); the average is taken over all simulator evaluations run by the algorithm during the calibration process. The last column measures the portion of the total runtime (in minutes) specifically taken to calculate the gradient during the linearization step. For the Assign-EKF algorithm, this just requires reading the empirical assignment matrix from DynaMIT; similarly, for the LimEKF algorithm, this just requires reading the average gradient that was estimated offline from a data file. For the proposed MM-EKF algorithm, this gradient estimation runtime is comprised of two main components—estimating the metamodel parameters (α_j^h, $\boldsymbol{\beta}_j^h$, γ_j^h) and solving the analytical traffic model at the current state estimate. Finally, for the FD-EKF algorithm, this gradient estimation runtime involves evaluating the simulator at every perturbation of the state vector to numerically calculate the gradient. Note that only one replication of the gradient estimation step is used for the FD-EKF results.

The most striking result shown in Table 5.5 is the runtime of the FD-EKF algorithm. The average total runtime to calibrate parameters for a single 15-minute time interval is 921 minutes, or over 15 hours, which is orders of magnitude larger than the runtime for any of the other online calibration algorithms. This is not a realistic runtime for using the algorithm in practice for a high-dimensional calibration problem comparable to the size of this case study. While this runtime can realistically be brought down with additional parallelization (the gradient estimation step for the FD-EKF algorithm is easily parallelizable) or better hardware, the vast difference in computational cost between the FD-EKF algorithm and the other algorithms should play a significant role for practitioners in deciding on an online calibration algorithm. Many practitioners may have limited resources as well, so the use of FD-EKF for a large-scale problem could prove to be infeasible. Each of the other four algorithms is able to estimate calibration parameters for the Singapore case study within the

allotted time frame of 15 minutes. The Assign-EKF algorithm is quite fast, with an average runtime of 1.8 minutes per time interval that is mostly comprised of evaluating the simulator once at the current state estimate, since the gradient calculation cost is negligible. The LimEKF algorithm is similarly fast due to there being no online cost to estimating the gradient, but it also relies on computing an average gradient matrix offline from the FD-EKF algorithm, so similar scrutiny is needed in implementing the algorithm for a high-dimensional problem. Interestingly, the GLS algorithm has the slowest average runtime of the benchmark algorithms, excluding FD-EKF, with an average across all three algorithm runs of 5.6 minutes. Still, it identifies parameter values well within the 15-minute time step. As the GLS algorithm does not require any simulator evaluations or gradient information, no runtime values are reported for those metrics.

The proposed MM-EKF algorithm has an average total runtime of 4.4 minutes, comfortably within the computational time budget and even faster than the GLS approach. The majority of this time is spent evaluating the simulator once at the current time estimate to add to the bank of simulator-evaluated points used to fit the metamodel. Note that the average simulator evaluation time for the MM-EKF algorithm is approximately three times that of the benchmark approaches—this is due to the simulator being evaluated in a time-independent sense with a fixed warm-up period of one hour, as outlined in Section 3.2.1, instead of just for the current 15-minute time interval. The gradient calculation step takes less than one minute on average; further computational cost analysis shows that about half of this runtime is used to estimate the metamodel parameters, while the other half is used to solve the analytical traffic model at the current state estimate. That is, the analytical traffic model can solve the system of nonlinear equations given by Equations 3.5 through 3.10 for the Singapore network in approximately 0.2 minutes. For the MM-EKF algorithm, the remaining computational budget per 15-minute time interval can be used to generate more simulator-evaluated points in the local neighborhood of the current state estimate to provide a better localized fit of the metamodel to the simulator. With additional parallelization, even more points could be evaluated

by the simulator. This may help improve the gradient approximation step of the proposed MM-EKF algorithm and lead to improved calibration accuracy.

Another performance metric for the computational cost of an online calibration algorithm is the number of times the simulator is evaluated per calibration interval, particularly as used by the gradient approximation method. For a simulation-based Dynamic Traffic Assignment system, evaluating the simulator during the online calibration process is by far the most computationally intensive task and thus consumes the largest proportion of the computational budget. This can be seen in the average simulator evaluation runtimes reported in Table 5.5. For the MM-EKF, Assign-EKF, and LimEKF algorithms, the simulator is only evaluated once per calibration interval (at the current state estimate) in Equation 2.15 of Algorithm 1. The GLS algorithm requires no evaluations of the simulator. On the other hand, for the one run of the FD-EKF algorithm reported in this case study, the simulator is evaluated an average of 6042 times per calibration interval, as averaged across all calibration intervals in the simulation period. While theoretically central finite differences requires $2n$ simulator evaluations, where n is the dimension of the state vector, the FD-EKF algorithm evaluates the simulator fewer than $2 \times 4050 = 8100$ times due to the non-negativity boundary constraints on O-D demand parameters. Still, the number of simulator evaluations required per interval by the FD-EKF algorithm is on the order of the dimension of the state vector of calibration parameters, a significant issue for high-dimensional calibration problems. As mentioned previously, it would be interesting to compare the proposed MM-EKF algorithm to other EKF algorithms from the literature that use different numerical gradient estimation techniques (i.e., simultaneous perturbation), both in terms of calibration accuracy and computational performance.

5.2.5 Impact of offline simulated points

There are several aspects of the proposed Metamodel EKF algorithm design that can be further explored with case studies, including different metamodel formulations, metamodel fitting techniques, and robustness to the quality of prior parameter estimates. The framework proposed in Chapters 2 and 3 of this thesis allows for

Algorithm	Run	counts		densities		O-D	
		RMSN	% improv	RMSN	% improv	RMSN	% improv
Baseline		0.143	-	0.364	-	1.172	-
MM-EKF with Offline Points	1	0.088	38.3	0.334	8.4	1.176	-0.3
	2	0.092	35.7	0.328	9.9	1.168	0.3
	3	0.090	36.9	0.327	10.3	1.184	-1.1
	Avg	0.090	37.0	0.330	9.5	1.176	-0.4
MM-EKF	1	0.082	42.7	0.329	9.8	1.387	-18.4
	2	0.086	40.1	0.322	11.7	1.518	-29.6
	3	0.085	41.0	0.335	8.0	1.532	-30.7
	Avg	0.084	41.3	0.329	9.9	1.479	-26.3

Table 5.6: Summary of estimation results for experiments on impact of offline simulator-evaluated points

considerable fine-tuning and experimentation with the algorithm depending on the problem structure and context. As an example, one final set of experiments for this case study show a direction for future refinement in developing the Metamodel EKF algorithm. The experiment results show some added benefit to the use of simulator-evaluated points generated during the offline phase in fitting the metamodel online, an algorithm step that has not been utilized in any of the case study results shown thus far.

A database of 100 simulator-evaluated points is generated offline prior to the online calibration process. Each of these 100 points, chosen from the space of possible parameter values (the set $0 \leq \mathbf{x}^h \leq \mathbf{x}^{\max}$ in the formulation of the O-D calibration problem defined by Equation 3.4) was created by adding an independent randomly drawn and normally distributed error term with an expectation of zero and a standard deviation of 20% of the prior O-D demand values. The MM-EKF algorithm is re-run three times for the Singapore expressway demand profile, though in these algorithm runs the online phase of the MM-EKF approach detailed in Algorithm 2 is initialized with the bank of 100 simulator-evaluated points. These points are used in each interval of the calibration process to help fit the metamodel to the simulator, in addition to the usual points generated during the online calibration process. Table 5.6 shows initial results for this analysis comparing estimated sensor counts, sensor densities, and O-D demand parameters for the new set of runs (denoted "MM-EKF with

Offline Points") compared to the previous MM-EKF algorithm runs from Tables 5.1 and 5.3. The format of the table is the same as for Table 4.2. While there is not a large difference in performance in fit to sensor counts or densities, the calibration runs with the database of 100 offline points do significantly improve in the RMSN metric for estimated O-D demand parameters—the significance of the improvement is confirmed with a one-sided Mann-Whitney test (p-value of 0.05). Whereas the MM-EKF algorithm with no offline simulator-evaluated points shows a worse performance than the baseline (average RMSN of 147.9%), the same MM-EKF algorithm starting with 100 offline points reports performance essentially equivalent to the baseline (average RMSN of 117.6%). Figure 5-16 shows error metrics for the estimated O-D parameters graphically in an hour-by-hour breakdown. In every hour after the initial hour of 6:00am to 7:00am, the version of the algorithm with a bank of 100 offline points for fitting the metamodel approximation (purple bar) has a lower average RMSN than the version with no offline points (orange bar). Although fit to the underlying O-D demand parameters is not the explicit objective of the online calibration problem, this set of preliminary results shows a statistically significant difference in estimation through the use of offline-generated simulator-evaluated points to fit the metamodel in the MM-EKF algorithm. It is possible that the additional points used to fit the metamodel parameters provide a better understanding of the global relationship between the simulator parameters and output. While this does not improve the local fit of the metamodel in estimating the gradient and fitting sensor measurements, perhaps this additional knowledge leads to a better estimation of the underlying parameters. A more detailed exploration of this aspect of the algorithm design, such as which contexts for generating these points has the biggest impact on calibration performance, is worth investigating in future research.

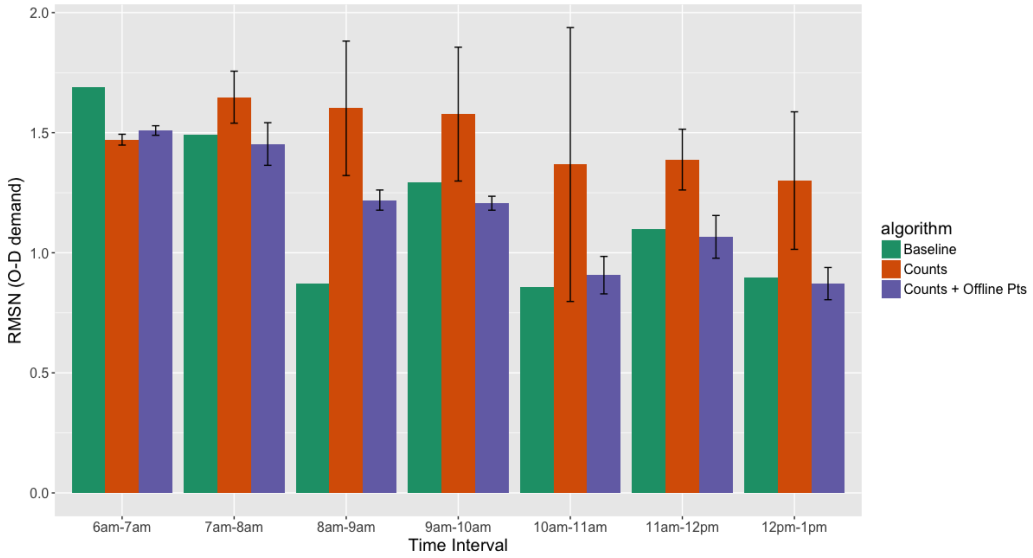

Figure 5-16: Impact of offline simulator-evaluated points on estimation of O-D parameters by the MM-EKF algorithm for Singapore case study

5.2.6 Analysis of calibrated measurements and O-D parameters

This section summarizes the performance of the proposed MM-EKF algorithm in estimating sensor count observations, in addition to estimation of other traffic state measurements like sensor densities and the underlying O-D demand parameters. In previous sections, the proposed algorithm has been analyzed in terms of calibration performance compared to benchmark online calibration algorithms and computational performance. Here, focus is brought back to the primary goal of the proposed MM-EKF algorithm, which is to tune simulator parameters so the simulator output replicates measurements observed in the field. The figures shown below illustrate these results for the Singapore network case study.

Figures 5-17 and 5-18 show the fit of the counts (in vehicles per 15 minute interval) and densities (in vehicles per mile per lane) estimated by the three runs of the MM-EKF algorithm, respectively, to the true sensor count and density measurements. The x-axis for both figures shows the "true" values of the field measurements, while the y-axis for both figures shows the corresponding values estimated by the MM-EKF algorithm. Each colored shape in the scatterplot represents a separate run of

the algorithm. A perfect fit of estimated measurements to true measurements, which would give a RMSN error of 0, is represented on the scatterplots by the blue "45-degree line" that corresponds to exact correlation. The fit for sensor counts in Figure 5-17 is very closely clustered around the 45-degree line, and does not show a large difference among the three runs of the MM-EKF algorithm. As reported in Table 5.1, the average RMSN values for the three runs are 8.2%, 8.6%, and 8.5%, all of which are below 10% indicating a good fit to observations. As illustrated, the proposed MM-EKF algorithm is able to identify O-D parameter values that produce simulator outputs replicating real-time field data from the Singapore network. In Figure 5-18 for estimated sensor densities, the fit between true measurements and MM-EKF estimated measurements is more diffuse, and it seems like there is a higher variance for larger density values, though the points are still generally clustered around the 45-degree line. While sensor densities were not provided as field data to the MM-EKF algorithm, it still does a reasonable job of replicating density measurements within the network.

It is worthwhile to note that for the Singapore network case study, the 4050 O-D demand parameters were calibrated only using sensor counts from 172 sensors in the network. In the literature review provided in Section 1.3, this falls on both the larger side of networks consider by the case study and the sparse side of field data available for use in online calibration. For their case studies on a slightly different representation of the Singapore expressway network, Prakash [52] and Zhang [73] use 357 and 650 sensor counts to calibrate 4121 O-D parameters, respectively. In discussing the limits of effective O-D matrix calibration using traffic counts, Marzano et al. [41] argue that sensor coverage as measured by the ratio of sensors to calibration parameters should be close to one for satisfactory O-D estimation results. They corroborate this claim with a set of experiments on large-scale synthetic networks. For this case study, the ratio of sensors to parameters is less than 5%. Considering this, the MM-EKF algorithm is able to show good performance in replicating real-time field observations from the network in an online calibration setting for the Singapore network case study.

Since the demand scenario used for the Singapore expressway network case study

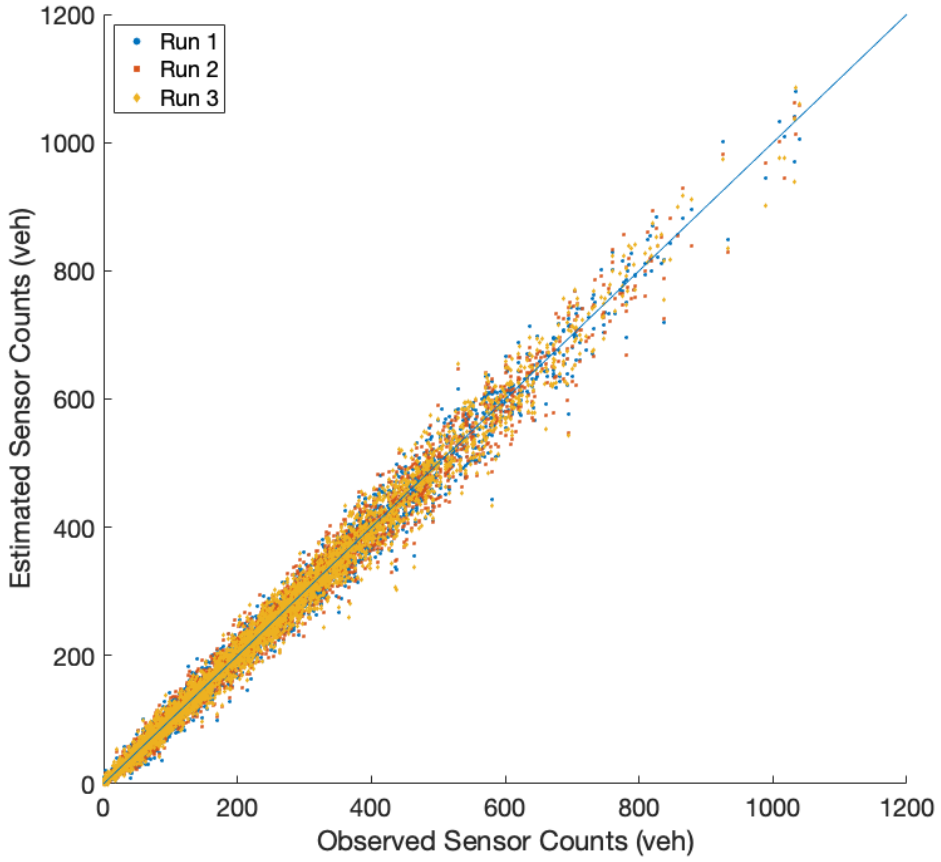

Figure 5-17: Scatterplot of field measurement sensor counts vs. MM-EKF estimated sensor counts for Singapore case study

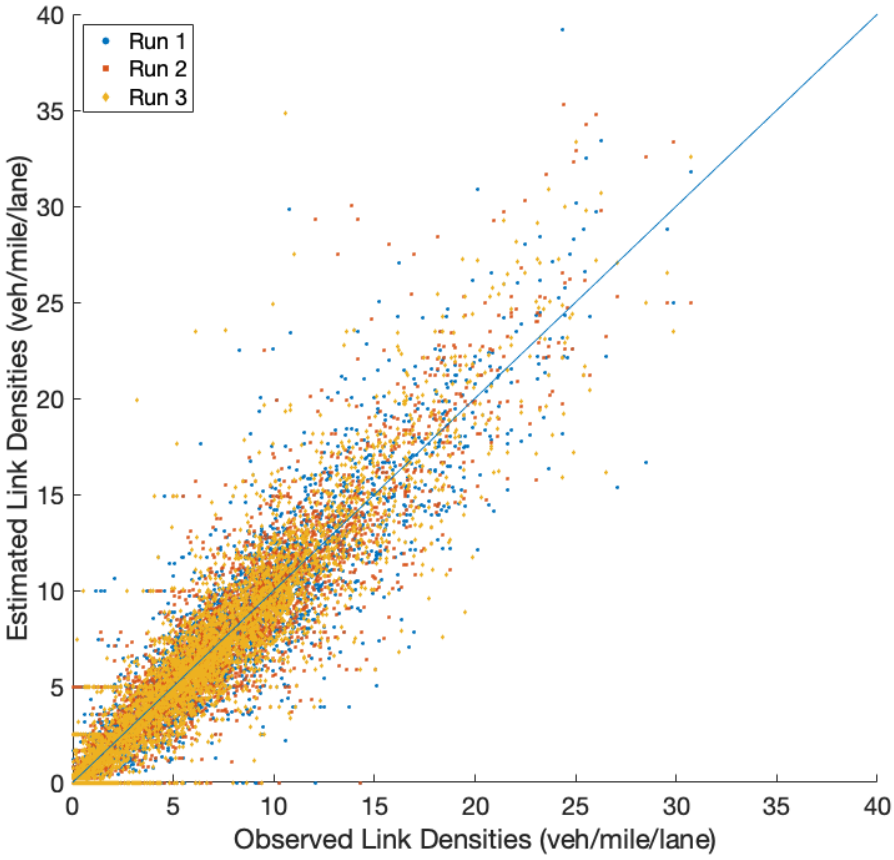

Figure 5-18: Scatterplot of field measurement sensor densities vs. MM-EKF estimated sensor densities for Singapore case study

is synthetic, it is also possible to compare the O-D demand parameters estimated by the MM-EKF algorithm with the "true" O-D demand parameters. Figures 5-19 and 5-20 make this comparison. Figure 5-19 shows a scatterplot of the true O-D demand parameters (in vehicles per 15 minute interval) on the x-axis and the O-D demand parameters estimated by the MM-EKF algorithm (in vehicles per 15 minute interval) on the y-axis. Each colored shape represents a separate run of the MM-EKF algorithm. From the plot, it seems the estimated O-D parameters are broadly clustered around the 45-degree line. The correlation coefficients of the relationship between estimated O-D parameters and true O-D parameters are 0.93 for the first run, 0.92 for the second run, and 0.92 for the third run, indicating a strongly linear fit. For the larger O-D demand parameter values, there does seem to be a slight bias in estimation, with the estimated values being larger than the corresponding true values.

Figure 5-20 gives a grouping of nine subplots that show true and estimated O-D demand parameters over time for the same subset of O-D pairs as in Figure 5-4. For each subplot (O-D pair), the x-axis plots the entire simulation period, and the y-axis plots the demand for the O-D pair in vehicles per 15 minutes. The blue solid line shows the true O-D demand over time, the red dashed line shows the prior O-D demand over time, and the remaining colors graph the three runs of the MM-EKF algorithm. Across the plotted subset of O-D pairs, the estimated parameters generally follow the same trends as the true O-D demand. The three different calibration runs are clustered closely together, which suggests some consistency among algorithm runs in the face of simulator stochasticity. Even though the general trends may be correct, the estimated parameter values differ noticeably from the true parameter values. This is not surprising given that many combinations of O-D demand parameter values can lead to the same sensor count measurements, but it does emphasize that the underlying parameter values may not be estimated accurately. As discussed, this can lead to poor predictive performance since the evolution of O-D demand will not be modeled accurately. In addition, the MM-EKF algorithm seems to weight the prior parameter values heavily, as estimated O-D demand plateaus at certain values before

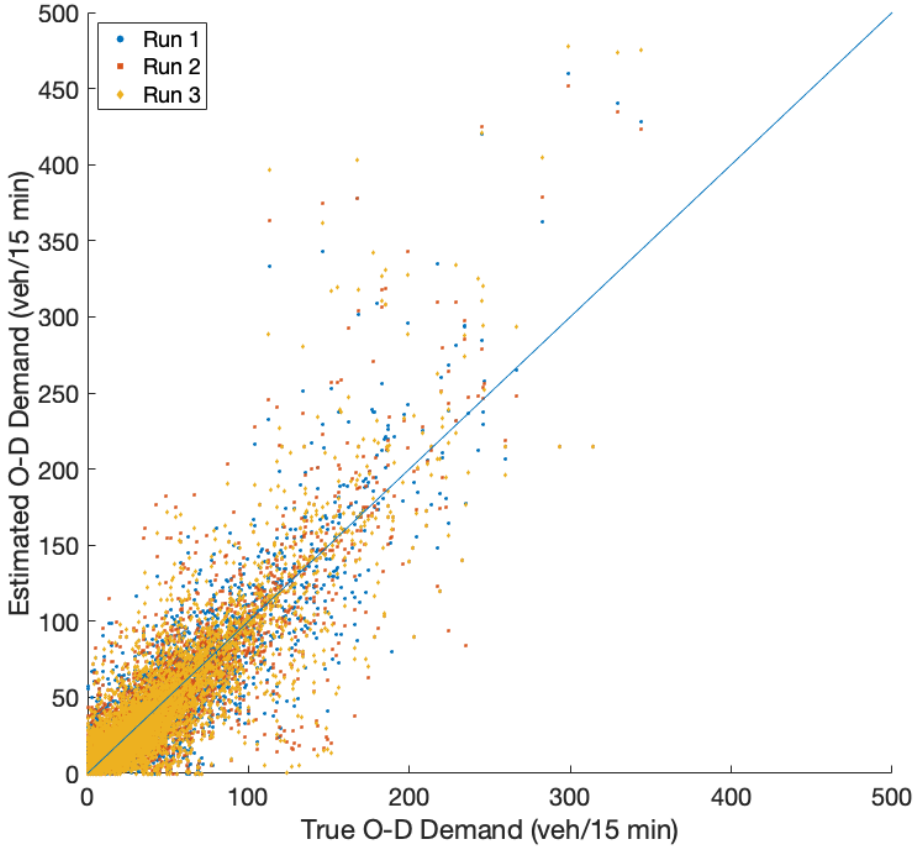

Figure 5-19: Scatterplot of true O-D demand parameters vs. MM-EKF estimated O-D demand parameters for Singapore case study

shifting whenever the prior demand changes for some of the O-D pairs. This can be observed particularly for O-D Pair 138 and 1668. This confirms the significant influence of the prior demand values on the MM-EKF calibration results; the quality of the O-D demand calibration is strongly related to the quality of the prior O-D demand parameters. Observations from these case study experiments may be used to re-tune the online calibration algorithm parameters (specifically the prior information weight parameter), perhaps to de-emphasize the influence of the prior demand on the resulting parameter estimates.

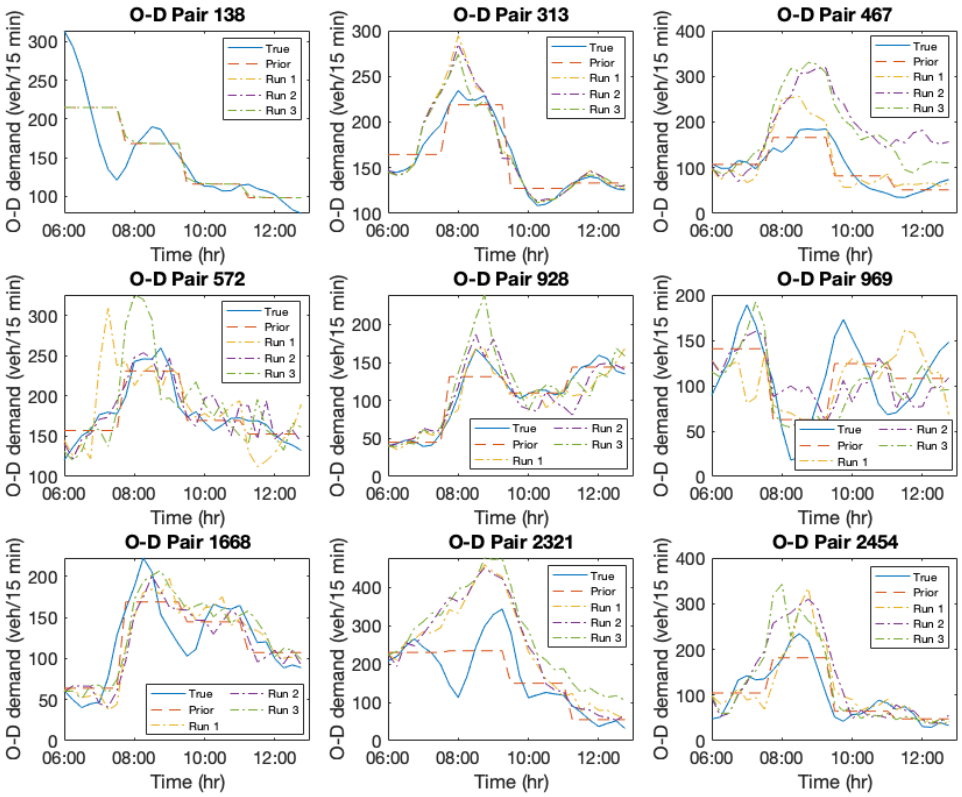

Figure 5-20: Estimated O-D demand parameters from MM-EKF online calibration for a subset of O-D pairs for Singapore network case study

5.3 Conclusion

In the application of the Metamodel EKF algorithm to a large-scale online calibration problem formulated on the Singapore expressway network, certain performance attributes of the proposed algorithm were highlighted. In particular, its feasibility was demonstrated on a real-world case study with relative improvement in estimation of field measurements over the baseline of prior values for calibration parameters. The computational advantages of using an analytically tractable metamodel in the gradient estimation step of the extended Kalman filter, as compared to numerical approximation methods, are shown for a large-scale problem. Limitations of the EKF modeling approach were demonstrated and discussed. The improvements seen in the Florian toy network case study did not fully translate to the larger real-world network, particularly in predictive performance, but further refinements of the MM-EKF algorithm implementation may potentially lead to better results. Through the case study, several avenues for further algorithm development are highlighted, which are discussed in more detail in the following chapter.

Chapter 6

Conclusion

In this thesis, a new algorithm for online calibration called the Metamodel EKF (MM-EKF) was proposed, developed, and then evaluated on several case studies. The general framework was laid out in Chapter 2, followed by extensive discussion of the algorithmic details and the flexibility of the approach in Chapter 3. An application of the algorithm to the online origin-destination (O-D) demand calibration problem was also presented. In Chapters 4 and 5, the proposed algorithm was validated on two different road networks—a small toy network called the Florian network and a large-scale urban road network of the major expressways and arterials of Singapore. The calibration accuracy and computational cost of the approach were assessed in comparison to other online calibration approaches from the literature. The feasibility for real-world applications, along with specific strengths, of the proposed algorithm were highlighted through experiments. In this concluding chapter, the major findings are summarized. Useful learnings from the algorithm development and implementation are discussed, as well as avenues for future exploration.

6.1 Major findings

6.1.1 Analytical traffic model for O-D demand

The case studies in the two previous chapters on the Florian and Singapore networks validated the analytical model presented in Section 3.3 for modeling the relationship between O-D demand and link attributes (counts, densities). The analytical model incorporates concepts from traffic flow theory like the fundamental diagram relationship and a multinomial route choice model within a system of nonlinear equations representing the traffic network. For the two case studies, fit of analytical model sensor counts to simulator outputs was very good with correlation coefficients of 0.999 and 0.984, respectively. In general, the fit for sensor densities also showed a strong linear relationship, though for some sensors in the case study networks the estimates differed from the simulator output by a constant scaling factor. The analytical traffic model provided a good estimate of the traffic state on the network as modeled by a stochastic traffic simulator, and proved to be both analytically tractable and scalable for a problem the size of the Singapore expressway network. For the large-scale urban network, the analytical model was able to be solved for a set of O-D demand flows in a matter of seconds. In addition, the MM-EKF algorithm was used successfully for online calibration in Section 4.3 for a demand scenario that showed high levels of congestion and rapid changes in demand with the help of the embedded analytical traffic model.

6.1.2 Metamodel extended Kalman filter algorithm

The main finding of this thesis is the development of a new algorithm for online calibration that embeds problem-specific structural information from an analytical traffic model in a general-purpose extended Kalman filter framework. The primary goal of the new algorithm is to improve the computational efficiency of the extended Kalman filter approach proposed by Antoniou [3] through the use of an analytical gradient calculation instead of computationally expensive numerical gradient approximation

methods. The algorithm is flexible in that it is able to jointly calibrate different types of simulator parameters and incorporate multiple sources of field measurements under one common framework. The main limitation of the approach is the need to construct an analytical model relating the calibration parameters to field measurements that is analytically differentiable. The Metamodel EKF algorithm was also shown to be more robust to the impact of simulator stochasticity than other general-purpose methods due to its avoidance of numerical gradient approximation methods. Finally, through the case study in Chapter 5, the proposed online calibration algorithm was shown to be tractable for high-dimensional calibration problems, and demonstrated better calibration accuracy and computational efficiency than other online calibration algorithms in the literature. The Metamodel EKF algorithm aims to fill the gap between general-purpose online calibration approaches and those that exploit problem structure.

6.1.3 Performance compared to benchmark algorithms

In validating the performance of the Metamodel EKF algorithm in Chapters 4 and 5, the proposed algorithm was compared to various benchmark algorithms from the literature in terms of both calibration accuracy and computational efficiency. Through the empirical case studies, advantages of the Metamodel EKF algorithm over each of the benchmark algorithms were demonstrated. Table 6.1 summarizes the main takeaways. Each row of the table outlines a desirable characteristic for online calibration, and the corresponding algorithms that demonstrated those characteristics in the case studies are marked. Only the MM-EKF algorithm demonstrates all of the desired characteristics listed in the table. The proposed algorithm is compared to three benchmarks that also utilize a state-space model formulation and extended Kalman filtering algorithm, and differ only in the method used for gradient approximation—(1) finite differences EKF, (2) limiting EKF, and (3) assignment matrix EKF. These are denoted in Table 6.1 as FD-EKF, LimEKF, and Assign-EKF, respectively. The generalized least-squares (GLS) algorithm is also used as a performance benchmark.

The MM-EKF algorithm, along with benchmarks FD-EKF and LimEKF, is able

	MM-EKF	FD-EKF	LimEKF	Assign-EKF	GLS
Calibrates parameters of all types	✓	✓	✓		
Accepts multiple types of field data	✓	✓	✓		
Analytical gradient calculation	✓				N/A
Computationally efficient for high-dimensional problems	✓		✓	✓	✓

Table 6.1: Comparison of the Metamodel EKF approach to benchmark online calibration algorithms

to handle the joint calibration of multiple parameter types (i.e., demand and supply model parameters) using multiple types of data. The advantage of being able to calibrate using multiple sources of field data is illustrated in the Florian network case study described in Section 4.3. The MM-EKF and FD-EKF algorithms calibrate O-D demand parameters to sensor counts and densities to varying outcomes. The MM-EKF algorithm is able to adjust the model parameters in response to a sudden increase in traffic congestion as observed in link densities. The Assign-EKF and GLS benchmarks are unable to accommodate density field measurements and thus miss the spike in demand, while the FD-EKF benchmark has difficulty incorporating the density observations into its calibration framework. Due to the impact of simulator stochasticity on the FD-EKF gradient approximation, the FD-EKF benchmark does not produce accurate estimates of the sensor count and density measurements, while the MM-EKF is more successful.

Among the online calibration algorithms, only the MM-EKF algorithm utilizes an analytical gradient calculation. The FD-EKF and LimEKF benchmarks rely on numerical gradient calculation techniques, while the Assign-EKF benchmark uses the empirical assignment matrix output by the traffic simulator. The GLS benchmark algorithm does not require a gradient. The analytical gradient used by the MM-EKF algorithm allows for incorporation of structural information through an analytical traffic model. Another impact of the analytical gradient is observed in the case study on the Singapore network in Chapter 5. The use of an analytical gradient significantly reduces the online computational cost of the extended Kalman filter algorithm framework for high-dimensional problems. The computationally expensive numerical gradient approximation used by FD-EKF prevents it from being tractable

for high-dimensional problems. The MM-EKF algorithm does not have the same scaling issues. The Assign-EKF and GLS benchmarks are also able to scale for high-dimensional problems, though only for the special case of O-D demand parameter calibration using traffic flow counts.

6.2 Limitations and insights

Through implementing the Metamodel EKF algorithm for the Florian toy network and Singapore expressway network case studies, several insights were gained about the process of adapting the MM-EKF algorithm to specific online calibration problems. A few of the complications and learnings are summarized in this section.

6.2.1 State-space modeling assumptions

In the large-scale Singapore network experiments, several difficulties were encountered related to the state-space modeling assumptions of the MM-EKF algorithm. The performance of the extended Kalman filter approach relies on several strong assumptions about the dynamics and evolution of the system being calibrated, from imposing a functional evolution of the state vector through the transition equation to expecting the random error terms to have a Gaussian distribution. For a real-world problem, these assumptions need to be examined, and decisions should be made about whether they are realistic enough or not. As with any model, divergence of the model from the actual network behavior leads to discrepancies in performance and for a dynamical system, these errors propagate through time. Inherent in this problem of model specification is the trade-off between model complexity and computational cost. A simpler state-space model, which may show larger divergence from reality, is less computationally costly than a more intricate modeling of the system. Both of these factors must be considered in formulating the state-space model, since both the calibration accuracy and computational efficiency of an algorithm are crucial in an online calibration context.

This is particularly true with regard to the transition equation in the state-space

model, as was seen in the empirical results of the Singapore network case study in Chapter 5. Transition equation modeling is especially important for the calibration of dynamic O-D flows, as they have a intricate spatial and temporal relationship with network observations. The Metamodel EKF approach in the case study used a simple random walk transition equation, which proved to be too reductive of a model of the evolution of dynamic O-D demand for the network and its effect on traffic flows. The largest impact of this overly simplified modeling assumption was seen in the poor predictive performance of the Kalman filtering algorithms. As discussed in Sections 1.3 and 5.2.2, a more detailed autoregressive transition equation can be used for the O-D demand calibration problem with a higher autoregressive degree. This modification is able to represent temporal delays in the relationship between O-D demand and link counts, improving the validity of the state-space model, but it requires extensive calibration to determine the best values for the autoregressive degree and autocorrelation coefficients. The augmented transition equation also increases the computational cost of the approach.

Another consideration in the state-space model formulation is how prior information is incorporated, and what effect that modeling assumption will have on the outcomes of the online calibration algorithm. For the Metamodel EKF algorithm, the state vector is defined in terms of the absolute parameter values; prior information is incorporated using direct measurement equations in the state-space model which allows for more sensitive tuning of the influence of the prior depending on its quality. On the other hand, more exploration into the use of deviations, as proposed by Ashok and Ben-Akiva [9], as the state may improve the legitimacy of the state-space modeling assumptions.

In considering these state-space model assumptions, a rigorous experimental design can help in testing various model formulations. Yet these adjustments require a sizable amount of historical data, a topic which is discussed next.

6.2.2 Practical considerations

In parallel with the considerations that must be taken in modeling a real-world scenario as a state-space model, there were a few practical issues that had to be addressed in the case studies. First, the EKF algorithm parameters (i.e., error covariance matrices, prior demand weight parameter) proved to be difficult to tune offline. Historical data is generally needed to tune these parameters, as well as a systematic method for finding optimal algorithm parameters. Overfitting can also be an issue in the calibration of the algorithm parameters. Several of these issues are addressed by Antoniou in Chapter 5 of his thesis [3], but rely on a considerable amount of historical data. For the Singapore network case study, there was a lack of offline historical data, so more ad hoc methods had to be used, and the parameter settings found can be further fine-tuned. Better calibration of these parameters may lead to improved performance in estimation and prediction.

A significant difficulty encountered during the Singapore case study was in creating synthetic true and prior demand scenarios for testing the online calibration algorithms. No real-world data was available for the network, and the obstacles in creating a realistic but challenging case study quickly became apparent. For the synthetic demand scenarios constructed for the Singapore network, it was difficult to create a set of prior demand parameters that modeled realistic demand on the network, but left enough room for improvement so that online calibration methods would be able to show significant change over the baseline of prior demand. For several demand scenarios that were tried and eventually for the demand scenario ultimately used in the case study, prior O-D demand values that deviated observably from the true O-D demand led to sensor measurements (counts, densities) that remained relatively similar. On the other hand, case studies with real-world data have their own set of complications, namely unknown true values for the parameters.

6.3 Further research directions

6.3.1 Metamodel development and refinement

While the case studies demonstrate one application of the Metamodel EKF algorithm to the O-D demand calibration problem and validate the performance of the algorithm in several contexts, many aspects of the proposed algorithm framework have not been fully explored and are worth looking into. Chapter 3 of the thesis lays much of the groundwork for these future research questions, specifically in developing and testing different variations in the metamodel formulation (i.e., different analytical models, higher degree polynomial functional component), procedures for fitting the metamodel to the simulator, and methods for generating simulator-evaluated points. The performance of these refinements of the MM-EKF algorithm should be demonstrated in empirical studies, both in terms of calibration accuracy and computational efficiency. Future real-world applications of the MM-EKF approach should also consider joint calibration of demand and supply parameters, as well as the incorporation of different sources of field data. In addition, there are several outstanding issues that might be addressed with extensions of the current MM-EKF framework. Some examples include modeling temporal dynamics explicitly within the metamodel formulation, how this could possibly be used to improve predictive performance, and an explicit approximation of the simulator gradient using the metamodel.

6.3.2 Applications of analytical traffic model

Given the promising validation results in Chapters 4 and 5 for the analytical traffic model developed in Section 3.3, there may be applications for the model in online calibration methods. Though the analytical model is effectively utilized by the Metamodel EKF algorithm to improve the computational efficiency of the baseline EKF approach, the observed effects of the embedded structural information in the Kalman filtering method are limited. The strong performance of the analytical model in validation experiments does not show that significant of an impact on the online cal-

ibration performance in the case studies, and there may be alternative methods that better exploit the analytical model. It seems like the added value of the analytical model may be hampered by the extended Kalman filtering framework, as the two approaches are somewhat at odds with one another. The analytical model provides a good high-level approximation of network attributes but the extended Kalman filter requires a good local approximation (of the gradient) in order to perform well. An interesting avenue for future research would be looking to develop an online calibration approach that takes advantage of the global properties of the analytical model, while remaining scalable for high-dimensional problems.

Appendix A

List of Notation

A.1 Online calibration formulation

$\mathbf{x}^h$	vector of calibration parameters for time period h;
$\mathbf{x}_a^h$	vector of prior calibration parameter values for time period h;
$\mathbf{y}^h$	vector of observed field measurements for time period h;
y_j^h	observed field measurement j for time period h;
$\hat{\mathbf{y}}(\cdot)$	vector of expected simulator estimates for measurements given input parameters;
$\hat{y}_j(\cdot)$	expected simulator estimate of measurement j given input parameters;
w	weight parameter for prior information;
n	dimension of parameter vector $\mathbf{x}^h$;
h	time period being calibrated;
Ω	set of constraints on the vector of calibration parameters;
$\mathcal{J}$	set of field measurements observed in real time.

A.2 State-space model

$\mathbf{x}^h$	state vector of calibration parameters for time interval h;
$\mathbf{y}^h$	measurement vector of observed field data for time interval h;
$\mathbf{u}^h$	Gaussian error term for transition equation for time interval h;
$\mathbf{Q}^h$	covariance matrix of error term in transition equation for time interval h;
$\mathbf{F}_{h+1}^q$	parameter of autoregressive transition equation relating the state at time q to the state at time $h+1$;
$\hat{\mathbf{y}}(\cdot)$	vector of expected simulator estimates for measurements;
$\mathbf{v}_1^h$	Gaussian error term for indirect measurement equation for time interval h;
$\mathbf{R}_1^h$	covariance matrix of error term in indirect measurement equation for time interval h;
$\mathbf{x}_a^h$	vector of prior calibration parameter values for time interval h;
$\mathbf{v}^h$	Gaussian error term for direct measurement equation for time interval h;
$\mathbf{R}^h$	covariance matrix of error term in direct measurement equation for time interval h;
h	time interval in state-space model;
p	degree of autoregressive transition equation;

A.3 Metamodel Extended Kalman Filter

$\mathbf{x}^0$	initial estimate of state vector of calibration parameters;
$\mathbf{P}^0$	initial covariance matrix of state vector $\mathbf{x}^0$;
$\mathbf{x}^{h\|h-1}$	estimate of state vector for time interval h based on measurements up to time $h-1$;
$\mathbf{P}^{h\|h-1}$	estimate of covariance matrix for time interval h based on measurements up to time $h-1$;
$\mathbf{x}^{h\|h}$	estimate of state vector for time interval h based on measurements up to time h;
$\mathbf{P}^{h\|h}$	estimate of covariance matrix for time interval h based on measurements up to time h;
$\mathbf{H}^h$	approximation of gradient at state estimate $\mathbf{x}^{h\|h-1}$;
$\mathbf{G}^h$	Kalman gain matrix for time interval h;
h	time interval in state-space model;

A.4 Metamodel approximation

$\mathbf{m}(\cdot)$	vector of metamodel estimates for measurements given input parameters;
$m_j(\cdot)$	metamodel estimate of measurement j given input parameters;
$\mathbf{q}(\cdot)$	vector of analytical traffic model estimates for measurements given input parameters;
$q_j(\cdot)$	analytical traffic model estimate of measurement j given input parameters;
$\phi(\cdot)$	functional component of metamodel formulation;
α_j^h	parameter of metamodel for measurement j at time interval h;
$\boldsymbol{\beta}_j^h$	parameter of metamodel for measurement j at time interval h;
γ_j^h	parameter of metamodel for measurement j at time interval h;
$\mathbf{x}_i$	calibration parameter candidate (i.e., simulator-evaluated point) from offline phase;
$\mathbf{x}_{h,i}$	calibration parameter candidate (i.e., simulator-evaluated point) from online phase;
δ_i^h	weight assigned to point $\mathbf{x}_i$ at time interval h during metamodel parameter estimation;
h	time period being calibrated;

A.5 Online O-D demand calibration

$\mathbf{x}^{\max}$ upper bound vector for O-D demand;

Endogenous variables of analytical model

d_s expected hourly demand for O-D pair s;
y_i expected hourly demand per lane for link i;
k_i expected density per lane for link i;
v_i expected (space-mean) speed for link i;
τ_i expected travel time for link i;
t_r expected travel time for route r;
$P(r)$ route choice probability for route r.

Exogenous variables of analytical model

k_i^{jam} jam density per lane of link i;
$v_i^{\max}$ maximum speed of link i;
q^{cap} lane flow capacity;
ℓ_i average lane length of link i;
n_i number of lanes of link i;
$\alpha_{1,i}, \alpha_{2,i}$ fundamental diagram parameters of link i;
θ travel time coefficient in the route choice model;
c scaling parameter common to all links;
$\mathcal{O}(r)$ O-D pair of route r;
$\mathcal{R}_1(i)$ set of routes that include link i;
$\mathcal{R}_2(s)$ set of routes of O-D pair s;
$\mathcal{L}(r)$ set of links of route r.

A.6 Kalman filter implementation for case studies

a parameter used to calculate initial covariance matrix $\mathbf{P}^0$ and transition equation random error covariance matrix $\mathbf{Q}^h$;
q_0 parameter used to calculate initial covariance matrix $\mathbf{P}^0$ and transition equation random error covariance matrix $\mathbf{Q}^h$;
b_1 parameter used to calculate indirect measurement equation random error covariance matrix $\mathbf{R}_1^h$;
$r_{0,1}$ parameter used to calculate indirect measurement equation random error covariance matrix $\mathbf{R}_1^h$;
b_2 parameter used to calculate direct measurement equation random error covariance matrix $\mathbf{R}_2^h$;
$r_{0,2}$ parameter used to calculate direct measurement equation random error covariance matrix $\mathbf{R}_2^h$;
w weight parameter of prior O-D demand;
$\mathbf{A}^h$ assignment matrix used by Assign-EKF benchmark algorithm for mapping O-D demand flows to link flows in time interval h;

Bibliography

[1] A. Allström, J. Ekström, D. Gundlegard, R. Ringdahl, C. Rydergren, A. M. Bayen, and A. D. Patire. Hybrid Approach for Short-term Traffic State and Travel Time Prediction on Highways. *Transportation Research Record*, 2554(1):60–68, 2016.

[2] A. Anand, G. Ramadurai, and L. Vanajakshi. Data Fusion-based Traffic Density Estimation and Prediction. *Journal of Intelligent Transportation Systems*, 18(4):367–378, 2014.

[3] C. Antoniou. *On-line Calibration for Dynamic Traffic Assignment*. PhD dissertation, Massachusetts Institute of Technology, Department of Civil and Environmental Engineering, September 2004.

[4] C. Antoniou, C. L. Azevedo, L. Lu, F. Pereira, and M. E. Ben-Akiva. W-SPSA in Practice: Approximation of Weight Matrices and Calibration of Traffic Simulation Models. *Transportation Research Part C*, 59:129–146, 2015.

[5] C. Antoniou, M. E. Ben-Akiva, and H. N. Koutsopoulos. Nonlinear Kalman Filtering Algorithms for On-line Calibration of Dynamic Traffic Assignment Models. *IEEE Transactions on Intelligent Transportation Systems*, 8(4):661–670, 2007.

[6] C. Antoniou, A. Kondyli, G. Lykogianni, and E. Gyftodimos. Exploratory Assessment of the Limiting Extended Kalman Filter Properties. *Transport and Telecommunication*, 14(1):1–12, 2013.

[7] C. Antoniou, H. N. Koutsopoulos, and G. Yannis. An Efficient Non-linear Kalman Filtering Algorithm using Simultaneous Perturbation and Applications in Traffic Estimation and Prediction. In *Proc. of the 2007 IEEE Intelligent Transportation Systems Conference*, pages 217–222, Seattle, 2007. IEEE.

[8] K. Ashok. *Estimation and Prediction of Time-dependent Origin-Destination Flows*. PhD dissertation, Massachusetts Institute of Technology, Department of Civil and Environmental Engineering, September 1996.

[9] K. Ashok and M. Ben-Akiva. Dynamic Origin-Destination Matrix Estimation and Prediction for Real-time Traffic Management Systems. In *Transportation and traffic theory*, number 12 in International Symposium on the Theory of Traffic Flow and Transportation, pages 465–484, Berkeley, 1993. Elsevier.

[10] K. Ashok and M. E. Ben-Akiva. Alternative Approaches for Real-time Estimation and Prediction of Time-dependent Origin-Destination Flows. *Transportation Science*, 34(1):21–36, 2000.

[11] K. Ashok and M. E. Ben-Akiva. Estimation and Prediction of Time-dependent Origin-Destination Flows with a Stochastic Mapping to Path Flows and Link Flows. *Transportation Science*, 36(2):184–198, 2002.

[12] V. Astarita, K. Er-Rafia, M. Florian, M. Mahut, and S. Velan. Comparison of Three Methods for Dynamic Network Loading. *Transportation Research Record*, 1771(1):179–190, 2001.

[13] J. Barceló, L. Montero, M. Bullejos, O. Serch, and C. Carmona. A Kalman Filter Approach for Exploiting Bluetooth Traffic Data when Estimating Time-dependent OD Matrices. *Journal of Intelligent Transportation Systems*, 17(2):123–141, 2013.

[14] N. Bekiaris-Liberis, C. Roncoli, and M. Papageorgiou. Highway Traffic State Estimation with Mixed Connected and Conventional Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3484–3497, 2016.

[15] M. Ben-Akiva, H. N. Koutsopoulos, C. Antoniou, and R. Balakrishna. Traffic Simulation with DynaMIT. In J Barceló, editor, *Fundamentals of Traffic Simulation*, pages 363–398. Springer, New York, 2010.

[16] M. Ben-Akiva, A. De Palma, and I. Kaysi. Dynamic Network Models and Driver Information Systems. *Transportation Research Part A*, 25(5):251–266, 1991.

[17] Y. Bie, T. Z. Qiu, C. Zhang, and C. Zhang. Introducing Weather Factor Modelling into Macro Traffic State Prediction. *Journal of Advanced Transportation*, 2017(1):1–15, 2017.

[18] M. Bierlaire and F. Crittin. An Efficient Algorithm for Real-time Estimation and Prediction of Dynamic OD Tables. *Operations Research*, 52(1):116–127, 2004.

[19] L. Cai, Z. Zhang, J. Yang, Y. Yu, T. Zhou, and J. Qin. A Noise-immune Kalman Filter for Short-term Traffic Flow Forecasting. *Physica A*, 536, 2019.

[20] S. Carrese, E. Cipriani, L. Mannini, and M. Nigro. Dynamic Demand Estimation and Prediction for Traffic Urban Networks Adopting New Data Sources. *Transportation Research Part C*, 81:83–98, 2017.

[21] E. Cascetta. Estimation of Trip Matrices from Traffic Counts and Survey Data: A Generalized Least Squares Estimator. *Transportation Research Part B*, 18(4–5):289–299, 1984.

[22] E. Cascetta, D. Inaudi, and G. Marquis. Dynamic Estimators of Origin-Destination Matrices using Traffic Counts. *Transportation Science*, 27(4):363–373, 1993.

[23] E. Cascetta, A. Papola, V. Marzano, F. Simonelli, and I. Vitiello. Quasi-dynamic Estimation of O-D Flows from Traffic Counts: Formulation, Statistical Validation and Performance Analysis on Real Data. *Transportation Research Part B*, 55:171–187, 2013.

[24] H. Chen and H. A. Rakha. Real-time Travel Time Prediction using Particle Filtering with a Non-explicit State-transition Model. *Transportation Research Part C*, 43(1):112–126, 2014.

[25] C. K. Chui and G. Chen. *Kalman Filtering with Real-time Applications*, volume 17 of *Springer Series in Information Sciences*, section 4, pages 49–66. Springer, New York, third edition, 1999.

[26] C. F. Daganzo. The Cell Transmission Model: A Dynamic Representation of Highway Traffic Consistent with the Hydrodynamic Theory. *Transportation Research Part B*, 28(4):269–287, 1994.

[27] T. Djukic, G. Flötteröd, H. van Lint, and S. Hoogendoorn. Efficient Real-time OD Matrix Estimation based on Principal Component Analysis. In *Proc. of the 2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 115–121, Anchorage, 2012. IEEE.

[28] A. Emami, M. Sarvi, and S. A. Bagloee. Using Kalman Filter Algorithm for Short-term Traffic Flow Prediction in a Connected Vehicle Environment. *Journal of Modern Transportation*, 27:222–232, 2019.

[29] J. Guo, W. Huang, and B. M. Williams. Adaptive Kalman Filter Approach for Stochastic Short-term Traffic Flow Rate Prediction and Uncertainty Quantification. *Transportation Research Part C*, 43(1):50–64, 2014.

[30] B. Hammit, R. James, and M. Ahmed. A Case for Online Traffic Simulation: Systematic Procedure to Calibrate Car-following Models using Vehicle Data. In *Proc. of the 2018 21st International Conference on Intelligent Transportation Systems*, pages 3785–3790, Maui, 2018. IEEE.

[31] H. Hashemi, K. F. Abdelghany, and A. F. Abdelghany. A Multi-agent Learning Approach for Online Calibration and Consistency Checking of Real-time Traffic Network Management Systems. *Transportmetrica B*, 5(3):364–384, 2017.

[32] D. Henclewood, W. Suh, M. Rodgers, M. Hunter, and R. Fujimoto. A Case for Real-time Calibration of Data-driven Microscopic Traffic Simulation Tools. In *Proc. of the 2012 Winter Simulation Conference*, pages 1–12, Berlin, 2012. IEEE.

[33] A. Hofleitner, R. Herring, and A. Bayen. Arterial Travel Time Forecast with Streaming Data: A Hybrid Approach of Flow Modeling and Machine Learning. *Transportation Research Part B*, 46(9):1097–1122, 2012.

[34] E. Huang. Algorithmic and Implementation Aspects of On-line Calibration of Dynamic Traffic Assignment. Master's thesis, Massachusetts Institute of Technology, Department of Civil and Environmental Engineering, June 2010.

[35] B. R. Hunt, E. J. Kostelich, and I. Szunyogh. Efficient Data Assimilation for Spatiotemporal Chaos: A Local Ensemble Transform Kalman Filter. *Physica D*, 230:112–126, 2007.

[36] N. Huynh, H. S. Mahmassani, and H. Tavana. Adaptive Speed Estimation using Transfer Function Models for Real-time Dynamic Traffic Assignment Operation. *Transportation Research Record*, 1783(1):55–65, 2002.

[37] D. Inaudi, E. Kroes, S. Manfredi, and S. Toffolo. The DYNA On-line Origin-Destination Estimation and Prediction Model. In *Proc. of the First World Congress on Applications of Transport Telematics and Intelligent Vehicle-Highway Systems*, number 3 in Towards an Intelligent Transport System, pages 1174–1180, Paris, 1994. Transportation Research Board.

[38] C. Lu and X. Zhou. Short-term Highway Traffic State Prediction using Structural State Space Models. *Journal of Intelligent Transportation Systems*, 18(3):309–322, 2014.

[39] Z. Lu, W. Rao, Y. Wu, L. Guo, and J. Sha. A Kalman Filter Approach to Dynamic OD Flow Estimation for Urban Road Networks using Multi-sensor Data. *Journal of Advanced Transportation*, 49:210–227, 2015.

[40] H. S. Mahmassani. Dynamic Network Traffic Assignment and Simulation Methodology for Advanced System Management Applications. *Networks and Spatial Economics*, 1(3):267–292, 2001.

[41] V. Marzano, A. Papola, and F. Simonelli. Limits and Perspectives of Effective O-D Matrix Correction using Traffic Counts. *Transportation Research Part C*, 17(2):120–132, 2009.

[42] V. Marzano, A. Papola, F. Simonelli, and M. Papageorgiou. A Kalman Filter for Quasi-dynamic O-D Flow Estimation and Updating. *IEEE Transactions on Intelligent Transportation Systems*, 19(11):3604–3612, 2018.

[43] L. Mihaylova, R. Boel, and A. Hegyi. Freeway Traffic Estimation within Particle Filtering Framework. *Automatica*, 43(2):290–300, 2007.

[44] L. Mussone and M. Matteucci. OD Matrices Network Estimation from Link Counts by Neural Networks. *Journal of Transportation Systems Engineering and Information Technology*, 13(4):84–92, 2013.

[45] C. Nanthawichit, T. Nakatsuji, and H. Suzuki. Application of Probe-vehicle Data for Real-time Traffic-state Estimation and Short-term Travel-time Prediction on a Freeway. *Transportation Research Record*, 1855(1):49–59, 2003.

[46] D. Ngoduy. Applicable Filtering Framework for Online Multiclass Freeway Network Estimation. *Physica A*, 387(2–3):599–616, 2008.

[47] C. Osorio and B. Atasoy. Efficient Simulation-based Toll Optimization for Large-scale Networks. Technical report, Massachusetts Institute of Technology, Cambridge, MA, 2017.

[48] C. C. Paige and M. A. Saunders. LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, 1982.

[49] T. L. Pan, A. Sumalee, R. X. Zhong, and N. Indra-payoong. Short-term Traffic State Prediction based on Temporal-spatial Correlation. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1242–1254, 2013.

[50] V. Papathanasopoulou, I. Markou, and C. Antoniou. Online Calibration for Microscopic Traffic Simulation and Dynamic Multi-step Prediction of Traffic Speed. *Transportation Research Part C*, 68:144–159, 2016.

[51] A. A. Prakash, R. Seshadri, C. Antoniou, F. C. Pereira, and M. E. Ben-Akiva. Reducing the Dimension of Online Calibration in Dynamic Traffic Assignment Systems. *Transportation Research Record*, 2667(1):96–107, 2017.

[52] A. A. Prakash, R. Seshadri, C. Antoniou, F. C. Pereira, and M. E. Ben-Akiva. Improving Scalability of Generic Online Calibration for Real-time Dynamic Traffic Assignment Systems. *Transportation Research Record*, 2672(48):79–92, 2018.

[53] X. Qin and H. S. Mahmassani. Adaptive Calibration of Dynamic SpeedâĂŞ-Density Relations for Online Network Traffic Estimation and Prediction Applications. *Transportation Research Record*, 1876:82–89, 2004.

[54] S. Sarkka. *Bayesian Filtering and Smoothing*, volume 3 of *IMS Textbooks*, section 4, pages 51–63. Cambridge University Press, Cambridge, UK, 2013.

[55] T. Singliar. *Machine Learning Solutions for Transportation Networks*. PhD dissertation, University of Pittsburgh, November 2008.

[56] J. C. Spall. Multivariate Stochastic Approximation using a Simultaneous Perturbation Gradient Approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992.

[57] *Degree of Urbanization in the United States from 1970 to 2018*, 2019 (accessed June 6, 2020). `https://www.statista.com/statistics/269967/urbanization-in-the-united-states`.

[58] S. Sun, C. Zhang, and G. Yu. A Bayesian Network Approach to Traffic Flow Forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):124–132, 2006.

[59] H. Tavana and H. S. Mahmassani. Estimation and Application of Dynamic Speed-Density Relations by using Transfer Function Models. *Transportation Research Record*, 1710(1):47–57, 2000.

[60] T. Toledo, T. Kolechkina, P. Wagner, B. Ciuffo, C. Azevedo, V. Marzano, and G. Flötteröd. Network Model Calibration Studies. In W. Daamen, C. Buisson, and S. P. Hoogendoorn, editors, *Traffic Simulation and Data: Validation Methods and Applications*, pages 141–162. CRC Press, Boca Raton, 2015.

[61] *United Nations Department of Economic and Social Affairs*, 2018 (accessed June 6, 2020). `https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html`.

[62] H. van Lint and C. van Hinsbergen. Short-term Traffic and Travel Time Prediction Models. In S. Kikuchi, editor, *Artificial Intelligence Applications to Critical Transportation Issues*, pages 22–41. Transportation Research Board, Washington, 2012.

[63] J. W. C. van Lint, S. P. Hoogendoorn, and H. J. van Zuylen. Accurate Freeway Travel Time Prediction with State-space Neural Networks under Missing Data. *Transportation Research Part C*, 13(5–6):347–369, 2005.

[64] C. Wang and Z. Ye. Traffic Flow Forecasting based on a Hybrid Model. *Journal of Intelligent Transportation Systems*, 20(5):428–437, 2016.

[65] 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*, 71:521–537, 2016.

[66] R. Wang, D. B. Work, and R. Sowers. Multiple Model Particle Filter for Traffic Estimation and Incident Detection. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3461–3470, 2016.

[67] Y. Wang and M. Papageorgiou. Real-time Freeway Traffic State Estimation based on Extended Kalman Filter: A General Approach. *Transportation Research Part B*, 39(2):141–167, 2005.

[68] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang. A Hybrid Deep Learning based Traffic Flow Prediction Method and Its Understanding. *Transportation Research Part C*, 90:166–180, 2018.

[69] H. Yang, P. J. Jin, B. Ran, D. Yang, Z. Duan, and L. He. Freeway Traffic State Estimation: A Lagrangian-space Kalman Filter Approach. *Journal of Intelligent Transportation Systems*, 23(6):525–540, 2019.

[70] S. Yang, J. Wu, and Y. Du. Ensemble Learning for Short-term Traffic Prediction based on Gradient Boosting Machine. *Journal of Sensors*, 2024:1–15, 2017.

[71] Y. J. Yu and M. Cho. A Short-term Prediction Model for Forecasting Traffic Information using Bayesian Network. In *Proc. of the 2008 International Conference on Convergence and Hybrid Information Technology*, pages 242–247, Busan, 2008. IEEE Computer Society.

[72] C. Zhang, C. Osorio, and G. Flötteröd. Efficient Calibration Techniques for Large-scale Traffic Simulators. *Transportation Research Part B*, 97:214–239, 2017.

[73] H. Zhang. *Online Calibration for Simulation-based Dynamic Traffic Assignment: Towards Large-scale and Real-time Performance*. PhD dissertation, Massachusetts Institute of Technology, Department of Civil and Environmental Engineering, September 2018.

[74] Y. Zhang and A. Haghani. A Gradient Boosting Method to Improve Travel Time Prediction. *Transportation Research Part C*, 58:308–324, 2015.

[75] X. Zhou and H. S. Mahmassani. A Structural State Space Model for Real-time Traffic Origin-Destination Demand Estimation and Prediction in a Day-to-day Learning Framework. *Transportation Research Part B*, 41:823–840, 2007.