On–line Calibration for Dynamic Traffic Assignment

by

Constantinos Antoniou

Diploma in Civil Engineering
National Technical University of Athens, Greece (1995)
Master of Science in Transportation
Massachusetts Institute of Technology (1997)

Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Transportation Systems

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2004

Author .....
Department of Civil and Environmental Engineering
August 20, 2004

Certified by .....
Moshe E. Ben-Akiva
Edmund K. Turner Professor
Department of Civil and Environmental Engineering
Thesis Supervisor

Certified by .....
Haris N. Koutsopoulos
Associate Professor
Department of Civil and Environmental Engineering, Northeastern University
Thesis Supervisor

Accepted by .....
Heidi Nepf
Chairman, Departmental Committee on Graduate Students

On–line Calibration for Dynamic Traffic Assignment

by

Constantinos Antoniou

Submitted to the Department of Civil and Environmental Engineering
on August 20, 2004, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Transportation Systems

Abstract

In this thesis, an on–line calibration approach for dynamic traffic assignment (DTA) that jointly estimates demand and supply parameters has been developed. The objective of on–line calibration is to introduce a systematic procedure that will use the available data to *steer* the model parameters to values closer to the *realized* ones. The approach is general and flexible and imposes no restrictions on the models, the parameters or the data that it can handle. The on–line calibration approach is formulated as a state–space model, comprising transition and measurement equations. *A priori* values provide direct measurements of the unknown parameters (such as origin–destination flows, segment capacities and traffic dynamics models' parameters), while surveillance information (for example, link counts, speeds and densities) is incorporated through indirect measurement equations. The state vector is defined in terms of deviations of the parameters and inputs that need to be calibrated from available estimates. Standard Kalman Filter theory does not apply to this formulation, as it is not linear. Therefore, three modified Kalman Filter methodologies are presented: Extended Kalman Filter (EKF), Limiting EKF, and Unscented Kalman Filter (UKF). A case study on a freeway network in Southampton, U.K., is used to demonstrate the feasibility of the approach, to verify the importance of on–line calibration, and to test the candidate algorithms. The empirical results from this application support the hypothesis that simultaneous on–line calibration of demand and supply parameters can improve the traffic estimation and prediction accuracy and show significant benefits (over the base case in which only the origin–destination flows are estimated on–line). In this application, the EKF has more desirable properties than the UKF. Furthermore, the Limiting EKF provides accuracy comparable to that of the *best* algorithm (EKF), but with computational complexity which is order(s) of magnitude lower than the other algorithms.

Thesis Supervisor: Moshe E. Ben-Akiva
Title: Edmund K. Turner Professor
Department of Civil and Environmental Engineering

Thesis Supervisor: Haris N. Koutsopoulos
Title: Associate Professor
Department of Civil and Environmental Engineering, Northeastern University

Acknowledgments

I would first like to thank my supervisors, Professor Moshe Ben-Akiva and Professor Haris N. Koutsopoulos, for their advice and guidance. I have learnt a lot from both of them in many fronts, extending well beyond academics.

I would also like to thank the members of my doctoral committee, Prof. Nigel Wilson, Dr. Kalidas Ashok, Dr. Michel Bierlaire, and Dr. Tomer Toledo, for valuable guidance, feedback and stimulating discussions.

Professor Cynthia Barnhart, Professor Joseph Sussman and Professor Ismail Chabini enriched my second MIT experience in different ways.

I am also grateful to my fellow researchers at the ITS lab (with some of which I spent way too much time). Rama has been a great office–mate and a good friend. With apologies to those that are not mentioned, I wish all the best to Akhil, Angus, Ashish, Bhanu, Charisma, Dan F., Dan M., Deepak, Emily, Joe, Kunal, Marge, Mark, Patrick, Song, Srini, and Yang. Prof. Ikki Kim and Dr. Frank Crittin have also contributed to the lab environment.

I would also like to thank the administrative staff at the Department of Civil and Environmental Engineering and the Center for Transportation and Logistics, and, in particular, Leanne Russell, Cynthia Stewart, and Sara Goplin for being always helpful.

I would like to thank Prof. Wilfred Recker from University of California, Irvine, and Dr. Paul McKenna from Lancaster University for providing the data used in this research.

Financial support from the New York State Department of Transportation, MIT Department of Civil and Environmental Engineering, and the Federal Highway Administration is gratefully acknowledged.

At last, but definitely not least, I would like to thank my family, for their endless support and constant encouragement, especially my parents Babi and Marika, my wife Mari–Elen, her parents, Ilia and Ester, and my brother Elia.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Contents

1.1 Motivation

Traffic congestion is a major problem in urban areas. It has a significant adverse economic impact through deterioration of mobility, safety and air quality. A recent study (FHWA, 2001) estimated that 32% of the daily travel in major US urban areas in 1997 occurred under congested traffic conditions. The annual cost of lost time and excess fuel consumption during congestion was estimated at $72 billion, over $900 per driver. These numbers represent a 300% increase from 1982.

Schrank and Lomax (2003) estimated that the cost of congestion continues to climb. 5.7 billion gallons of wasted fuel and 3.5 billion hours of lost productivity resulting from traffic congestion in 2001 cost the nation $69.5 billion, $4.5 billion more than the previous year. The extra time needed for rush hour travel has tripled over two decades. The national average *Travel Time Index* for 2001 was 1.39 (meaning a rush hour trip took 39 percent longer than a non-rush hour trip). The national average in 1982 was only 1.13.

As a result, the importance of better management of the road network to efficiently utilize existing capacity is increasing. To that end, many urban areas build and operate modern Traffic Management Centers (TMCs), which perform several functions, including collection and warehousing of real-time traffic data, and utilization of this data for various dynamic traffic control and route guidance applications. These applications require traffic models that provide, in real-time, estimation and prediction of traffic conditions. The complexity of transportation systems often dictates the use of detailed simulation-based Dynamic Traffic Assignment (DTA) models (Ben-Akiva et al., 1991, 2002; Mahmassani, 2001) for that purpose.

These complex models often involve a large number of parameters and inputs that need to be calibrated. In most cases, the approach to this problem has been to perform off–line calibration of the simulation models using a database of historic information. The calibrated parameter values are then used in the on–line simulations. The calibrated model parameters therefore represent average conditions over the period represented in the data. Models that were calibrated this way may produce satisfactory results in off–line evaluation studies, which are concerned with the expected performance of various traffic management strategies.

However, this may not be the case in real-time applications, which are concerned with the system performance on the given day. If the model calibrated off-line is used without adjustment, the system is not sensitive to the variability of the traffic conditions between days, which are the result of variations in the parameters of the system, such as weather and surface conditions. Such variations may cause traffic conditions to differ significantly from the average values. Thus, the predictive power of the simulation model may be significantly reduced. To overcome this problem, real-time data can be used to re-calibrate and adjust the model parameters on-line, so that prevailing traffic conditions can be captured more accurately. The wealth of information included in the off-line values can be incorporated into this process by using them as a priori estimates.

1.2 Dynamic traffic assignment framework

Dynamic Traffic Assignment (DTA) systems typically reside at Traffic Management Centers (TMC) and can support both planning and real–time applications. Planning applications may include the off–line evaluation of incident management strategies, the evaluation of alternative traffic signal and ramp meter operation strategies and the generation of evacuation and rescue plans for emergencies (e.g. natural disasters) that could affect the traffic network. Real–time applications make use of the traffic prediction capabilities of DTA systems and may include on–line evaluation of guidance and control strategies, real–time incident management and control, support of real–time emergency response efforts and optimization of the operation of TMCs through the provision of real–time predictions.

The on–line calibration approach presented in this thesis is relevant both in the planning and the real–time contexts. Special emphasis is given to the real–time problem, however, since runtime is important (therefore a practical formulation and efficient solution approaches are important) and it combines estimation and prediction capabilities.

Real–time DTA systems are also often referred to as *traffic estimation and prediction systems* (the two terms will be used interchangeably in this document). An overview of the state-of-the-art Dynamic Traffic Assignment framework is shown in Figure 1-1.

Real–time DTA systems typically comprise two main functions (Ben-Akiva et al., 2002):

- State estimation; and

- Prediction-based information generation

During the *state estimation* phase, real–time information is combined with historical data to capture the current traffic conditions in the network. Detailed traffic information that is obtained through the instrumented portions of the network is used to infer the conditions in the parts of the network for which no real-time information is available. This is achieved through an iterative simulation of demand-supply interaction designed to reproduce real-time observations from the surveillance system.

The role of the *prediction-based information generation process* is to generate unbiased and consistent traffic information for dissemination to travelers. Information based on predicted network conditions (i.e. anticipatory information) is likely to be more effective than information based on current traffic conditions because it accounts for the evolution of traffic conditions over time which is what travelers will experience. A detailed treatment of the demand-supply interactions within a state-of-the-art DTA system can be found in Ben-Akiva et al. (2002).

One of the key components of dynamic traffic assignment is the Origin-Destination (OD) estimation and prediction process (Ashok and Ben-Akiva, 2000). OD estimation combines historical and real-time information to obtain dynamic –i.e. time-dependent– demand matrices. Furthermore, OD prediction exploits estimated behavioral patterns to anticipate the short-term evolution of demand (Antoniou et al., 1997). Based on the predicted demand, it is possible to generate and evaluate response strategies and generate anticipatory guidance (Bottom, 2000).

The supply simulator is usually based on high–level (mesoscopic or macroscopic) models that represent traffic dynamics using speed-density relationships, kinematic representation of traffic elements of queueing theory, etc. Such models are used in real-time applications, e.g. DynaMIT (Ben-Akiva et al., 2002) and DYNASMART (Mahmassani, 2001). Proper use of these models requires calibration of a number of parameters.

Clearly, the speed–density relationship may depend on location–specific parameters, such as type of facility, number of lanes, lane width, slope, surroundings, as well as temporal variations, i.e. it may vary by season, day of the week, or even time of day, reflecting different driving behaviors (e.g. experienced drivers during commute periods). Off-line cal-

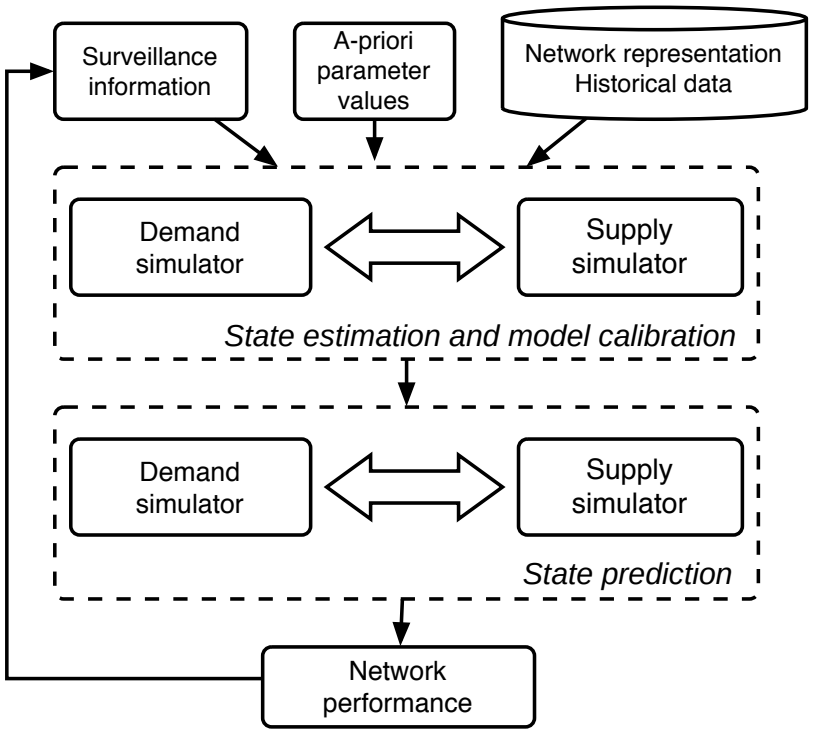

Figure 1-1: Dynamic traffic assignment framework overview

ibration could, in principle, deal with these situations, through the generation of a historical database of different speed-density relationships, categorized by the conditions. Based on the prevailing conditions, the "appropriate" relationship could then be retrieved and used.

However, traffic dynamics also depend on factors that cannot always be anticipated, such as weather conditions, incidents, unscheduled maintenance work, and even when they can be predicted, it would be impractical to calibrate traffic dynamics models for each possible scenario. Minor incidents (such a car slowing down in the break–down lane) that are not reported or captured otherwise in the system may also impact the traffic dynamics.

The output capacity of the network elements is another important parameter determining the traffic dynamics. Average values could in general be obtained during an off-line calibration phase. However, capacities are affected by several phenomena (including weather and lighting conditions, traffic composition, etc) and may therefore change as prevailing conditions change.

1.3 Problem definition and thesis objective

DTA systems combine aggregate and disaggregate models for the estimation and prediction of traffic conditions. Surveillance data reflecting the traffic conditions in parts of the network are fused with historical information to infer an estimate of the complete state of the network. Based on this state estimate, predictions can be made; the accuracy of these predictions naturally depends on the quality of the state estimates.

The models in a DTA system involve a number of parameters that need to be calibrated. These parameters are typically calibrated off–line, based on available archived data. The output of the off–line calibration is typically a single value for each parameter. However, these parameters represent random variables. Therefore, a proper off–line calibration procedure would succeed in determining the *mean* of the distribution of these random variables. More elaborate off–line calibration procedures could result in multiple parameter values (e.g. for different time periods, traffic or weather conditions). In effect these values would capture the mean of different parameter distributions, based on the prevailing conditions.

However, the *realized* values of these parameters will typically vary from the mean (for example, due to variations in the traffic mix or changes in the weather conditions). The use of the mean value (instead of the *realized*, unknown value) would introduce a discrepancy between the simulated and observed traffic conditions, even if other inputs were perfect and the models captured the true behavior of the system.

The *realized* values of the model parameters depend on the prevailing conditions. The surveillance information used as input for the models is a proxy for the prevailing conditions. Therefore, the same surveillance data could be used to *steer* the (off–line calibrated) values of the model parameters in the appropriate direction. Intuitively, using parameter values closer to the *realized* ones should lead to better estimates and predictions.

Therefore, the surveillance data can be used to systematically re–calibrate the model parameters on–line, during every interval. The most recent (and therefore most relevant) surveillance information would thus be used to *update* the off–line calibrated values of the model parameters, which are based on a large amount of information, reflecting a range of situations but not necessarily the time interval of interest. The synergistic nature of the on–line and the off–line calibration processes becomes clear. The on–line calibration relies

on and exploits the results of the off–line calibration. The two procedures are inherently complementary and when combined can improve the accuracy of DTA systems.

The objective of this thesis is to develop an on–line calibration approach for dynamic traffic assignment. The approach should be general and flexible, yet targeted at the particular characteristics of the application. Furthermore, it should be expressed in a practical form, that can be implemented and made operational.

1.4 Literature review

While the available literature on the topic of on–line calibration is limited, on–line calibration depends on off–line calibration and —to some extent— the two techniques follow the same general principles, albeit with important differences.

This section starts with a presentation of the state–of–the–art in off–line calibration, followed by a review of prior on–line calibration research. System–level approaches are presented first, followed by research focused on individual components.

1.4.1 Off–line calibration

In the context of dynamic traffic assignment, off–line calibration addresses the following problem: *Given a set of initial values for various parameters and aggregate measures of flows, speeds and densities at sensor locations, determine the OD flows, route choice parameters, capacities and speed-density relationships, so that the error between the simulated output and observed values is minimized.*

Balakrishna (2002) formulated the off–line calibration framework as a large optimization problem with the final objective of matching simulated and observed quantities:

$$\min_{\boldsymbol{\beta},\boldsymbol{\gamma},\mathbf{x}_{\mathfrak{p}}} \left\| \mathbf{M}^{\mathtt{sim}} - \mathbf{M}^{\mathtt{obs}} \right\| \tag{1.1}$$

where $\boldsymbol{\beta}$ represents the route choice parameters, $\boldsymbol{\gamma}$ represents the parameters in the supply simulator, $\mathbf{x}_{\mathfrak{p}}$ are the OD flows departing their origin during interval $\mathfrak{p}$, $\mathbf{M}^{\mathtt{obs}}$ are the observed measurements and $\mathbf{M}^{\mathtt{sim}}$ are their simulated counterparts. The objective func-

tion minimizes the discrepancy between various simulated and observed measurements $\mathbf{M}$. Flows, speeds and densities can be used as measurements. The simulated quantities are obtained through the following steps. Equation 1.2 forms the basic OD estimation step in the calibration framework, and is itself an optimization problem with a two-part objective function:

$$\mathbf{x}_h = \arg\min \left[\mathbf{F}_1(\mathbf{x}_h, \mathbf{x}_h^a) + \mathbf{F}_2 \left(\sum_{p=h-p'}^{h} \mathbf{a}_h^p \mathbf{x}_p, \mathbf{y}_h \right) \right] \tag{1.2}$$

Function $\mathbf{F}_1$ measures the Euclidian distance of the estimated flows $\mathbf{x}_h$ from their a priori values $\mathbf{x}_h^a$, while $\mathbf{F}_2$ measures the distance of the measured counts $\mathbf{y}_h$ from their simulated counterparts. The simulated flows can be represented in the following convenient linear form:

$$\mathbf{y}_h^{sim} = \sum_{p=h-p'}^{h} \mathbf{a}_h^p \mathbf{x}_p \tag{1.3}$$

where $\mathbf{a}_h^p$ is the assignment matrix mapping OD flows departing their origin during time interval p onto sensor counts measured during time interval h, and p' is the number of intervals required for the largest trip in the network. A thorough treatment of the assignment matrix can be found in Ashok and Ben-Akiva (2002).

The assignment matrices $\mathbf{a}_h^p$ required by the OD estimation module are outputs of the dynamic network loading model, and are functions of the as yet unknown OD flows, the equilibrium travel times on each link (tt_l^{eq}), the route choice parameters $\boldsymbol{\beta}$ and the supply-side parameters $\boldsymbol{\gamma}$:

$$\mathbf{a}_h^p = g\left(\mathbf{x}_p, \boldsymbol{\beta}, \boldsymbol{\gamma}, tt_l^{eq}\right) \tag{1.4}$$

Finally, the equilibrium travel times are themselves a function of the route choice parameters $\boldsymbol{\beta}$, supply-side parameters $\boldsymbol{\gamma}$, and OD flows $\mathbf{x}_p$:

$$tt_l^{eq} = h\left(\boldsymbol{\beta}, \boldsymbol{\gamma}, \mathbf{x}_p\right) \tag{1.5}$$

Equations 1.2 to 1.5 capture the fixed-point nature of the calibration problem. An iterative approach is therefore used to converge to a consistent calibration of the parameters and "equilibrium" conditions (Sundaram, 2002).

Balakrishna (2002) presents a methodology to jointly calibrate the OD estimation and prediction and driver route choice models within a DTA system using several days of traffic sensor data. The parameters to be calibrated include a database of time-varying historical OD flows, variance–covariance matrices associated with measurement errors, a set of autoregressive matrices that capture the spatial and temporal inter–dependence of OD flows, and the route choice model parameters. Issues involved in calibrating route choice models in the absence of disaggregate data are identified, and an iterative framework for jointly estimating the parameters of the O-D estimation and route choice models is proposed.

Kunde (2002) presents a three–stage methodology for the off–line calibration of mesoscopic flow propagation models (commonly used in DTA systems). Calibration is carried out in a sequential manner at increasing levels of aggregation. Speed–density relationship parameters and segment capacities are calibrated. Balakrishna et al. (2004) present an iterative off–line calibration approach that integrates demand and supply calibration. Furthermore, a review of the off–line calibration literature can be found at Balakrishna (2002) and Kunde (2002).

1.4.2 On–line calibration

System–level approaches

The topic of on-line calibration of traffic simulation models has received only limited attention in the literature. Doan et al. (1999) outline a framework for periodic adjustments to a traffic management simulation model in order to maintain an internal representation of the traffic network consistent with that of the actual network. Errors accumulate from time period to time period. Furthermore, errors propagate from one model to another. The authors categorize the error sources as demand estimation, path estimation, traffic propagation, internal traffic model structure, and on–line data observation and propose a system of on–line and off–line adjustment modules. Doan et al. recognize the importance of considering the effects of all models (demand and supply) during the on-line model refinement

process but they do not propose any specific approaches in that direction.

Doan et al. (1999) propose a reactive traffic propagation adjustment module. Inconsistencies between simulated and measured densities would be detected through this module and model parameters (in particular speeds) would be adjusted in real-time to correct them. The module is formulated as a PID (proportional, integral and derivative) controller. The impact of the on–line adjustment module is demonstrated through case simulations using synthetic data. The conclusion is that the simulator performs significantly better with the built–in reactive traffic propagation adjustment module.

A similar approach is proposed in Peeta and Bulusu (1999), where consistency is sought in terms of minimizing the deviations of the predicted time–dependent path flows from the corresponding actual flows.

He et al. (1999) develop a combined off–line and on–line calibration process to adjust the analytical dynamic traffic model's output to be consistent with real–world traffic conditions by periodically detecting inconsistencies between model outputs and real–world data, and actuating the correction model to correct the errors. The authors attempt to list the major sources of error in a DTA system, but then arbitrarily consider a subset of them: dynamic link travel time functions, route choice, and flow propagation models. Notably absent from the calibration is OD estimation.

He et al. consider a modified Greenshields' model (Greenshields, 1935) to explain dynamic travel time variations on freeway links, and split the travel times on arterials into a cruise time component and a delay component (capturing queuing at intersections). The suggested calibration approach aims to minimize the "distance" between the analytically computed travel times and those computed using speeds measured by detectors. However, the problem of using time–mean speed to compute travel times is not addressed. All authors agree that for computations involving mean speeds to be theoretically correct, it is necessary to ensure that one has measured space mean speed, rather than time mean speed (Hall, 1997).

The authors note that due to data and computational requirements, calibration of the route choice model should take place off–line. Finally, a capacity reduction factor and a proportional adjustment of travel times are used for the calibration of the flow propagation model. The authors propose an iterative (heuristic) approach that sequentially considers the three components until convergence. The convergence criterion is somewhat vague

("[The model parameters] are modified and adjusted until the model outputs represent the real–world conditions reasonably well"). Results from a simple test network are provided.

In the remainder of this section we present approaches that deal with the on–line adjustment of a subset of the parameters in a DTA system.

Supply parameters

van Arem and van der Vlist (1992) developed an on–line procedure for the estimation of current capacity at a motorway cross–section. The procedure is based on the combination of an on–line estimation of a "current" fundamental diagram with a maximum occupancy that may be achieved under free–flow conditions. The method is based on two assumptions. First, it is assumed that there exists a "current" fundamental diagram which depends on prevailing conditions. A method for establishing such fundamental diagrams based on on–line measurements of flow, occupancy and speed is presented. The second assumption is that the capacity can be estimated using this fundamental diagram and the notion of "maximum" occupancy. The capacity is estimated by substituting the current maximum occupancy into the current fundamental diagram.

Tavana and Mahmassani (2000) use transfer function methods (bivariate time series models) to estimate dynamic speed–density relations from typical detector data. The parameters are estimated using the past history of speed–density data; no predetermined parameters or shape for the model are assumed. The method is based on time series analysis, using density as a leading indicator. The resulting model is a descriptive rather than behavioral model to estimate speed and subsequently to predict its value for future time intervals.

The objective of identifying a transfer function model is to determine the appropriate form of the model and initial values of its parameters. Deviation of speed from its static equilibrium value is used as the output of the dynamic system. To make the input and output stationary, both speed and density are differenced once. The method can also be used when an equilibrium speed–density relation is not used. The performance of the approach is illustrated in both cases with encouraging results.

Huynh et al. (2002) extend the work of Tavana and Mahmassani (2000) by incorporating the transfer function model into a DTA simulation–based framework. Furthermore, the es-

timation of speeds using the transfer function model is implemented as an adaptive process, where the model parameters are updated on–line based on the prevailing traffic conditions. A nonlinear least squares optimization procedure is also incorporated into the DTA system to enable the estimation of the transfer function model parameters on–line.

Results from simulation–based experiments confirmed that the adaptive model outperforms the non–adaptive model. Sensitivity analysis on the updating frequency found that the most favorable model performance was obtained when an updating frequency of 10 minutes was used with a time series spanning 45 minutes of data. Furthermore, initial findings suggest that a transfer function model calibrated from one detector site can be used at another site (without a detector). The scope of this study, however, was limited to updating speeds on a single link using synthetic data; therefore the model was still not validated with real data.

Qin and Mahmassani (2004) address these shortcomings by evaluating the same model with actual sensor data from several links of the Irvine, CA, network. In this paper, determination of system input and output is derived from the higher–order continuum model. From the numerical results, the performance and the robustness of the transfer function model is in general found to be superior to the static modified Greenshields model. Specifically, the adaptive calibration outperforms the non–adaptive strategy and increasing the number of links to which the transfer function model is applied improves the speed estimation.

Wang and Papageorgiou (2004) present a general approach to the real-time estimation of the complete traffic state in freeway stretches. They use a stochastic macroscopic traffic flow model, and formulate it as a state-space model, which they solve using an Extended Kalman Filter. The formulation allows dynamic tracking of time-varying model parameters by including them as state variables to be estimated. A random walk is used as the transition equations for the model parameters.

With the on–line model parameter estimation activated, the overall estimation performance is found to be quite satisfactory after an initial warm–up period of up to 1.5 hours. Wang and Papageorgiou (2004) state that even a single "wrong" parameter leads to estimation bias in all segment variables. Furthermore, they state that while estimating all three parameters (free–flow speed, critical density, exponent) results in the best results, estimation of two parameters (free–flow speed and critical density or exponent, while fixing the third to its original value) does not cause significant performance degradation. The results also indicate

that the traffic state estimation is not sensitive to the initial values of the model parameters except for the warm–up period.

Demand parameters

Ashok and Ben-Akiva (1993), Ashok (1996), and Ashok and Ben-Akiva (2000) formulated the real–time OD estimation and prediction problem as a state–space model and solved it using a Kalman Filtering algorithm. The recursive solution approach has computational advantages and is therefore amenable to real–time application. One interesting characteristic of this approach is the use of *deviations* of OD flows (instead of the OD flows themselves) as variables. The use of deviations incorporates the wealth of structural information about spatial and temporal relationships between OD flows contained in the historical estimates into the OD estimation framework. The real–time OD estimation and prediction framework has been implemented in the DynaMIT DTA system (Antoniou et al., 1997; Ben-Akiva et al., 2002). An efficient solution algorithm for the OD estimation problem has been presented by Bierlaire and Crittin (2004).

1.4.3 Conclusion

The problem of on–line calibration of DTA systems has received some attention in the literature. Most existing methodologies, however, impose serious constraints and make restrictive assumptions. In particular, the components of a DTA system are considered in a sequential approach and iterative/heuristic approaches are proposed to estimate the appropriate parameters on–line.

Individual approaches for the on–line calibration of subsets of the parameters have also been developed. Such approaches update only a subset of the parameters in a DTA system. Therefore, all error or uncertainty is attributed to one source, which is unrealistic.

DTA systems capture complex demand and supply interactions (Ben-Akiva et al., 2002). Therefore, a calibration approach that would consider subsets of these parameters sequentially (while keeping the other parameters fixed) would not provide optimal results. Instead, an approach is needed that jointly estimates demand and supply parameters simultaneously and captures these interactions, thus ensuring consistency between the estimated parame-

ters.

1.5 Thesis contributions

This research makes several concrete contributions to the state–of–the–art. Specifically:

- A comprehensive framework for the on–line calibration of a Dynamic Traffic Assignment system is developed. The on–line calibration approach is expressed in a compact form with the following features:

  - Integrates the OD estimation and supply parameter estimation problems into a single formulation. Demand and supply parameters are thus estimated jointly. Demand–supply interactions can thus be captured in a consistent way.

  - Is generic and applicable to any DTA system. In particular, the on–line calibration approach does not make any assumptions on the specific models that comprise the DTA system. Therefore, it is applicable to systems with very different characteristics (e.g. analytical versus simulation–based, or microscopic versus macroscopic).

  - Has flexible data requirements. The approach can easily incorporate all available surveillance information —for example Automated Vehicle Identification (AVI) or probe vehicle data.

- New solution algorithms are applied to the problem. Besides the Extended Kalman Filter (EKF), which is a well–established algorithm for dealing with non–linear state–space models, the Limiting EKF and the Unscented Kalman Filter (UKF) are considered.

  - The accuracy achieved by the Limiting EKF is comparable to the EKF, but the computational cost is order(s) of magnitudes lower. The Limiting EKF makes the on–line calibration approach computationally feasible.

1.6 Thesis outline

The remainder of this document is organized as follows. Chapter 2 presents the overall on–line calibration approach. In Chapter 3 the on–line calibration approach is formulated as a state–space model. Solution approaches are presented in Chapter 4. Practical data and computational considerations are addressed in Chapter 5. Results from a case study are discussed in Chapter 6, while conclusions and directions for further research are outlined in Chapter 7.

Chapter 2

On–line calibration approach

Contents

The on–line calibration approach is presented in this chapter. An overview of the methodology is presented first, focusing on the inputs and outputs and placing the on–line calibration approach into the general DTA context. A general problem formulation is then presented, followed by the definition of the measurement equations.

2.1 Overview of methodology

Data for dynamic traffic assignment come from diverse sources. Surveillance data is fused with historical information to estimate and —subsequently— predict traffic conditions. The models that estimate and predict traffic conditions also require a number of parameters. Typically, these parameters are calibrated off–line. The same parameter values are then used irrespective of the prevailing conditions.

An incremental improvement is the off–line calibration of model parameters for different conditions. In this way, a library of model parameters is generated, from which the most appropriate can be selected and used. Furthermore, off–line re-calibrations may be used to incorporate new archived surveillance information and recent operational experience, thus refining the previously generated model parameters.

The objective of the on–line calibration approach is to introduce a systematic procedure that will use the available data to *steer* the model parameters to values closer to the true ones. In other words, the aim of the on–line calibration procedure is to obtain those model parameter values that will minimize the discrepancy between the observed measurements and their simulated counterparts (when these parameters are used as inputs to the models).

The inputs and outputs of the on–line calibration module are outlined in Figure 2-1 and discussed in the following paragraphs.

2.1.1 Inputs

The on–line calibration component exploits all information that is available within a traffic estimation and prediction system:

- Historical information describing the transportation system;
- *A priori* values of the model parameters;

- Surveillance data capturing the prevailing traffic conditions.

Historical information is the foundation for the development of a transportation model. Historical information capturing the supply side of the modeled transportation system may include the network geometry, traffic control settings and layout of the surveillance system. It should be noted that this information does indeed change over time, as —for example— a two–way street becomes one–way, a new traffic signal is added (or an existing one is reconfigured), or additional surveillance devices are deployed. In order to account for such changes, this information may be updated periodically. The term "historical" reflects the time frame of these updates, which is very different from the —much shorter— intervals at which dynamic surveillance information becomes available and the system operates.

***A priori* information** on the value of the model inputs and parameters is generally also available (usually from the output of an off–line calibration phase). Time–dependent demand (in the form of OD matrices) is one key input. Supply–side parameters include segment capacities and parameters of the speed–density relationships. Error variance–covariance matrices, and autoregressive fractions capturing the temporal evolution of these parameters may also be available.

Information on the prevailing traffic conditions becomes available in the form of **surveillance data.** The quality (and quantity) of this information is of paramount importance, as this effectively captures the prevailing conditions that the system is trying to match. The on–line calibration process aims to *steer* the model parameters towards values that –when used along with the other inputs– will minimize the discrepancy between the estimated (and predicted) conditions and these surveillance measurements.

The most common source of surveillance information are sensors, providing counts, speeds, and densities (point measurements). While several technologies have been introduced (microwave/acoustic sensors, cameras, etc), by far the most common is still inductive loop detectors.

Recent advances in sensor and telecommunication technologies have enabled additional sources of information, often collectively called Automatic Vehicle Identification (AVI) systems. Through the use of such systems it is possible to obtain very detailed information about a fraction of the vehicles (since usually additional equipment is required for a vehicle

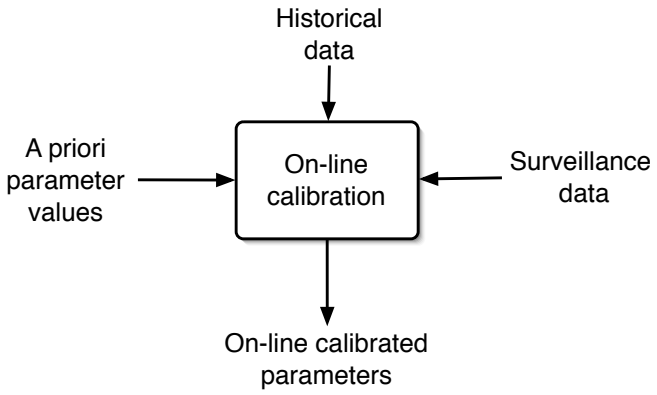

Figure 2-1: Overview of inputs and outputs

to be detected). This information often includes point–to–point travel time information, subpath flow counts and path choice fractions. Furthermore, due to the characteristics of the enabling technologies, AVI measurements have very high accuracy (Mouskos et al., 1998). However, since the information is obtained from a sample of the traffic, the final measurements will likely be biased (Hellinga and Fu, 2002).

2.1.2 Outputs

As has already been mentioned, the output of the on–line calibration module is a set of parameter values that —when used as input for the traffic estimation and prediction framework— minimizes the discrepancy between the simulated (estimated and predicted) traffic conditions and the observed traffic conditions. These parameters are a subset of the parameters used by the traffic estimation and prediction system and would exclude any parameters that are not expected to change from time–interval to time–interval in response to prevailing conditions. Using the DTA framework presented in Figure 1-1 as a reference, a discussion of the parameters of interest in the context of the on–line calibration follows.

The demand simulator combines two main components, the OD estimation (and prediction) and the behavioral models. The **OD flows** capture the variability of the demand to be loaded onto the network (from the historical demand). This variability may be a result of information provided to the drivers, day–to–day demand fluctuations, or intra–day demand variations. Therefore, OD flows are a key component that needs to be adjusted on–line. As a matter of fact, the OD estimation and prediction component of existing DTA systems

(Antoniou, 1997; Ben-Akiva et al., 2002) can be considered as a partial on–line calibration approach. Other parameters of the OD estimation and prediction process (such as the autoregressive fractions and the error covariance matrices) are not expected to vary in the short term, and therefore need not be subjected to on–line calibration. Instead, they can be captured by the off–line calibration (and periodic off–line re–calibrations, as needed).

Regarding the **behavioral model parameters**, it is noted that, while the behaviors of the drivers change, the underlying behavioral parameters do not change from interval to interval. The behavioral parameters that capture the driver's decisions are combined with the perceived conditions to lead to actual decisions. It is recognized, however, that slow changes of these behavioral parameters are possible (for example in response to changes in the socioeconomic characteristics of the population, learning patterns). Such changes can be captured through periodic off–line re–calibrations.

The supply simulator in DTA systems (e.g. DynaMIT (Ben-Akiva et al., 2002) and DYNASMART (University of Maryland, 2003)) is typically mesoscopic (or macroscopic) in nature. Higher-level (mesoscopic or macroscopic) simulation models represent traffic dynamics using speed-density relationships, fluid representation of traffic flow, elements of queueing theory, etc. **Supply model parameters** in this level include the speed–density relationship parameters and segment capacities.

2.1.3 The relation with off–line calibration

The scope of the on–line calibration is not to duplicate or substitute for the off–line calibration process. Instead, the two processes are complementary and synergistic in nature.

One of the first steps in a DTA system deployment is the off–line calibration process. During that process, large amounts of data are analyzed and used for the estimation of *a priori* values of the model inputs. During the on–line calibration, on the other hand, a small amount of data is used to refine these estimates. This data, however, is the most relevant, as it is capturing the prevailing traffic conditions.

On–line calibration needs to consider all prior information (condensed during the off–line calibration into the estimated parameter values) and exploit the limited, fresh surveillance information to determine the direction towards which parameter values for the immediate

simulation runs should be steered.

In other words, the aim of the on–line calibration is to use the off–line calibrated parameter values as starting points and perform a local optimization step towards the unobserved *true* values.

2.2 Problem formulation

The on–line calibration problem can be formulated in several ways. In this section, we present a general formulation that casts the problem as a non–linear optimization problem.

2.2.1 Preliminary definitions

Consider an analysis period of length $\mathcal{T}$ divided into equal intervals $h = 1, 2, \ldots, N$ of size T. Consider also a transportation network represented as a directed graph that includes a set of consecutively numbered nodes $\mathcal{N}$ and a set of numbered links $\mathcal{L}$. Each link includes one or more segments, all of which belong to a set of numbered segments $\mathcal{G}$. The network is assumed to have n_L links, n_g segments and n_{OD} OD pairs. Furthermore, it is assumed that n_l of the n_L links are equipped with surveillance sensors.

Let $\mathcal{S}(\boldsymbol{\pi})$ be a simulation–based traffic estimation and prediction system, where $\boldsymbol{\pi}$ is the vector of model parameters and inputs that need to be calibrated for time interval h. Let $\boldsymbol{\pi}^a$ denote *a priori* values of the parameters $\boldsymbol{\pi}$, for example their off–line calibrated values. Let $\mathbf{M}_h$ denote a vector of *true* traffic conditions for time interval h (e.g. link flow counts, subpath flow counts, point–to–point travel times, route–choice fractions), $\mathbf{M}_h^o$ be a vector of observed traffic measurements for time interval h and $\mathbf{M}_h^s$ be their simulated counterparts (obtained as output from the simulator $\mathcal{S}$).

2.2.2 Available information

Available information (Section 2.1.1) is associated with the unknown parameter values through measurement equations. *A priori* values of the model parameters provide **direct** measurements of the unknown parameters. Surveillance information, on the other hand, can be used to formulate **indirect** measurement equations, where the output of the simulator

model $\mathcal{S}$ (when the unknown set of parameter values is used as input) would match the surveillance information.

Direct measurements

By definition, a direct measurement provides a preliminary estimate of a parameter. Within the context of on–line calibration a preliminary estimate of the parameters is provided by the off–line calibration. Therefore, the vector of off–line calibrated parameter values $\boldsymbol{\pi}_{\mathfrak{h}}^{\mathfrak{a}}$ can be used as an *a priori* estimate of the *true* parameter vector $\boldsymbol{\pi}_{\mathfrak{h}}$.

The *a priori* values of the input parameters can be expressed as a function of the "true" parameters:

$$\boldsymbol{\pi}_{\mathfrak{h}}^{\mathfrak{a}} = \boldsymbol{\pi}_{\mathfrak{h}} + \boldsymbol{\epsilon}_{\mathfrak{h}}^{\mathfrak{a}} \tag{2.1}$$

where $\boldsymbol{\epsilon}_{\mathfrak{h}}^{\mathfrak{a}}$ is a vector of random error terms.

Direct measurements of some OD flows could also be available from advanced surveillance technologies, such as Automated Vehicle Identification (AVI) systems or probe vehicles. Such technologies allow the tracking of equipped vehicles as they move through the network, thus obtaining detailed surveillance information (based on a sample of the population). Under certain conditions, i.e. that the vehicles can be detected close to their origin and their destination, it is possible to infer direct measurements of OD flows (Antoniou et al., 2004). Such information could easily be incorporated as additional direct measurements.

Indirect measurements

Practically any type of traffic measurements can be used as indirect measurement equations. An indirect equation links the observed traffic measurements with their simulated counterparts when a particular set of parameters is used as input. In the general case, modeled trips last longer than one interval. Therefore, simulated trajectories of vehicles are impacted by the traffic conditions during previous intervals (and consequently by the model parameters used during these intervals). The simulated traffic measurements during

time interval h can therefore be represented as:

$$\mathbf{M}_h^s = \mathcal{S}(\boldsymbol{\pi}_h, \boldsymbol{\pi}_{h-1}, \ldots, \boldsymbol{\pi}_{h-p}) = \mathcal{S}(\boldsymbol{\Pi}_h) \tag{2.2}$$

where $\mathcal{S}$ is a mapping of the input parameters onto the measurements (representing the simulation model), p is the number of intervals required for the longest trip in the network, and $\boldsymbol{\Pi}_h = \boldsymbol{\pi}_h, \boldsymbol{\pi}_{h-1}, \ldots, \boldsymbol{\pi}_{h-p}$ is an augmented vector of parameters.

The relationship between the observed and the simulated measurements can then be written as follows:

$$\mathbf{M}_h^o = \mathbf{M}_h^s + \boldsymbol{\epsilon}_h^o \tag{2.3}$$

where $\boldsymbol{\epsilon}_h^o = \boldsymbol{\epsilon}_h^f + \boldsymbol{\epsilon}_h^s + \boldsymbol{\epsilon}_h^m$ is a compound observation error comprising three error sources:

- $\boldsymbol{\epsilon}_h^f$ captures structural errors (due to the inexactness of the simulation models),
- $\boldsymbol{\epsilon}_h^s$ captures simulation errors (e.g. sampling and numerical errors), and
- $\boldsymbol{\epsilon}_h^m$ captures measurement errors.

As it is not possible to distinguish between these three error components, however, they will be treated together. Furthermore, it is assumed that $\boldsymbol{\epsilon}_h^o$ is independent from the error vector $\boldsymbol{\epsilon}_h^a$ introduced in Equation 2.1.

2.2.3 The objective function

The on-line calibration problem can be formulated as a minimization problem where the objective function aims to jointly minimize the following components:

- $\boldsymbol{\epsilon}_h^o$: deviation of simulated traffic conditions $\mathbf{M}_h^s$ from the respective observed measurements $\mathbf{M}_h^o$, and
- $\boldsymbol{\epsilon}_h^a$: deviation of a set of parameters and inputs $\boldsymbol{\pi}_h$ (over which the optimization is performed) from their *a priori* values $\boldsymbol{\pi}_h^a$.

The objective function could then be expressed as:

$$\min_{\boldsymbol{\pi}_h} \quad \left[\mathcal{N}_1(\boldsymbol{\epsilon}_h^o) + \mathcal{N}_2(\boldsymbol{\epsilon}_h^a)\right] \tag{2.4}$$

where $\mathcal{N}_i(\cdot)$ are appropriate functions measuring the magnitude of the errors. For example, $\mathcal{N}_i(\cdot)$ may be the Euclidian norm.

Substituting the expressions for the error terms from Equations 2.1 and 2.3, the objective function can be restated as:

$$\min_{\boldsymbol{\pi}_h} \quad \left[\mathcal{N}_1(\mathbf{M}_h^o - \mathbf{M}_h^s) + \mathcal{N}_2(\boldsymbol{\pi}_h^a - \boldsymbol{\pi}_h)\right] \tag{2.5}$$

The above formulation can be made operational in a number of different ways, depending on the assumptions regarding the nature of the various error terms and the functional forms of $\mathcal{N}_i(\cdot)$. The various formulations may lead to different solution approaches with different convergence and computational properties. For example, if $\boldsymbol{\epsilon}_a$ and $\boldsymbol{\epsilon}_o$ are assumed to be normally distributed the formulation reduces to the following generalized least squares (GLS) problem:

$$\begin{aligned}\min_{\boldsymbol{\pi}_h} \quad \big[&(\mathbf{M}_h^o - \mathbf{M}_h^s)' \ \mathbf{W}^{-1}(\mathbf{M}_h^o - \mathbf{M}_h^s) + \\ &+(\boldsymbol{\pi}_h^a - \boldsymbol{\pi}_h)' \ \mathbf{V}^{-1}(\boldsymbol{\pi}_h^a - \boldsymbol{\pi}_h)\big]\end{aligned} \tag{2.6}$$

where $\mathbf{W}$ and $\mathbf{V}$ are the variance–covariance matrices of the measurements and *a priori* values, respectively.

The solution $\boldsymbol{\pi}_h^*$ to this optimization problem can then be obtained from:

$$\begin{aligned}\boldsymbol{\pi}_h^* = \arg\min_{\boldsymbol{\pi}_h} \quad \big[&(\mathbf{M}_h^o - \mathbf{M}_h^s)' \ \mathbf{W}^{-1}(\mathbf{M}_h^o - \mathbf{M}_h^s) + \\ &+(\boldsymbol{\pi}_h^a - \boldsymbol{\pi}_h)' \ \mathbf{V}^{-1}(\boldsymbol{\pi}_h^a - \boldsymbol{\pi}_h)\big]\end{aligned} \tag{2.7}$$

In an on–line application, however, this formulation would be impractical since the problem needs to be solved at every time interval, with all the information on previous time intervals (because of the temporal correlations between the errors). However, it does lend itself to be re–stated as a state–space model, which can then be solved efficiently using recursive methods such as Kalman Filtering techniques.

The on–line calibration approach can also be solved using other algorithms for non–linear systems of equations. A particularly suitable algorithm has been recently developed (Crittin,

2003; Crittin and Bierlaire, 2003) as a generalization of secant methods. The proposed class of methods calibrates a linear model based on several previous iterates. The difference with existing approaches is that the linear model to interpolate the function is not imposed. Instead, the linear model which is as close as possible to the nonlinear function (in the least–squares sense) is identified.

2.3 Conclusion

The on–line calibration approach has been presented in the context of a state–of–the–art traffic estimation and prediction system. A problem formulation has been developed, considering available information, and a discussion on possible operationalization approaches has been presented.

A reference should be made to the topic of over–fitting, which is an inherent concern in parameter estimation. The parameter calibration should be such that not only improves the estimation accuracy but –more importantly– maintains (and improves) the forecasting power of the model. A detailed treatment of this topic is presented in Chapter 6.

In Chapter 3 the problem is formulated as a nonlinear state–space model. Approaches for solving nonlinear state–space models are presented in Chapter 4.

Chapter 3

State–space formulation

Contents

In the previous chapter, the on–line calibration approach has been developed as a dynamic system and approaches to operationalize it have been discussed. A classical technique for dealing with dynamic systems is *state–space* modeling. In this section, the on–line calibration approach presented in Chapter 2 is formulated as a state–space model, comprising:

- Transition equations that capture the evolution of the state vector over time, and
- Measurement equations that capture the mapping of the state vector on the measurements.

Given that state–space models have been extensively studied and efficient algorithms have been developed to solve them, this formulation will lead us naturally to Chapter 4 where solution approaches are discussed.

The first step in developing a state–space model is to define the state vector (Section 3.1). In this context, the parameters and inputs that need to be calibrated define the state. Measurement equations have already been presented (Section 2.2.2). Transition equations are developed in Section 3.2.

A reformulation of the problem in terms of *deviations* is presented in Section 3.3.

3.1 State vector

The concept of the *state* (or *state vector*) is fundamental in the description of a state–space model. The state vector $\mathbf{x}_{\mathtt{h}}$ is defined as the minimal set of data that is sufficient to uniquely describe the dynamic behavior of the system at time interval $\mathtt{h}$ (the assumption of a *discrete*, stochastic, dynamic system is made).

Within the framework of state–space models, the state vector includes the parameters $\boldsymbol{\pi}_{\mathtt{h}}$ that need to be calibrated during time interval $\mathtt{h}$. Referring to the discussion in Section 2.1.2, the main parameters for the on–line calibration problem are:

- OD flows,
- Speed–density relationship parameters, and
- Segment capacities.

Let $\mathbf{x}_h$ be a vector of OD flows departing their origins during interval h, $\mathbf{p}_h$ be a vector holding the values of parameters of the speed–density relationship models during interval h, and $\mathbf{c}_h$ be a vector of segment capacities for interval h. Then the state vector can be represented by:

$$\boldsymbol{\pi}_h = [\mathbf{x}_h \quad \mathbf{p}_h \quad \mathbf{c}_h]^T = [\mathbf{x}_h \quad \boldsymbol{\gamma}_h]^T \tag{3.1}$$

where $\boldsymbol{\gamma}_h = [\mathbf{p}_h \quad \mathbf{c}_h]^T$ succinctly represents the supply models' parameters and the superscript T indicates transpose.

The dimension of the state vector is an important attribute of the model, largely governing the computational properties of any solution approach. The dimension of the state vector is the sum of the number of OD pairs, the total number of speed–density relationship parameters and the number of segment capacities.

The number of OD pairs n_{OD} in realistic applications is usually in the hundreds and can be in the thousands. Similarly, the number of segments n_g (for which output capacities need to be determined) can range from the hundreds to the thousands (depending on the extent of the network, as well as other modeling assumptions).

The speed–density model depends on a number of parameters $\mathbf{p}$. For example, a typical formulation is:

$$u = u_f \left[1 - \left(\frac{\max(0, K - K_{min})}{K_{jam}}\right)^{\beta}\right]^{\alpha} \tag{3.2}$$

The parameter vector in this case includes free–flow speed (u_f), minimum and jam density (k_{min} and k_{jam}, respectively) and two exponents (α and β).

A number of parameters n_p (for example, for the functional form presented in Equation 3.2: $n_p = 5$) would need to be calibrated for each segment, resulting to a total of $(n_p \times n_g)$ parameters.

Considering the n_{OD} OD flows, the n_g capacities and the $(n_p \times n_g)$ speed–density relationship parameters, the total dimension n_s of the state vector would then be given by:

$$n_s = n_{OD} + (n_p + 1) \times n_g \tag{3.3}$$

3.2 Transition equations

Transition equations capture the evolution of the state vector over time. A typical formulation for the transition equation relates the state during a given interval to a series of states from previous intervals.

A general formulation of such a transition equation would be:

$$\boldsymbol{\pi}_{k+1} = \mathcal{T}\left(\boldsymbol{\pi}_k, \boldsymbol{\pi}_{k-1}, \ldots, \boldsymbol{\pi}_{k-p}\right) + \boldsymbol{\eta}'_k \tag{3.4}$$

where $\mathcal{T}$ is a function capturing the dependence of the parameter vector $\boldsymbol{\pi}_{k+1}$ during interval $k+1$ on the values of the parameter vector during the past several intervals, p is the number of past parameter vectors that are considered, and $\boldsymbol{\eta}'_k$ is a vector of random error terms.

A common approach to the representation of transition equations is the use of autoregressive processes. Expressed as an autoregressive function, the transition equation can be written in matrix form as follows:

$$\boldsymbol{\pi}_{k+1} = \sum_{q=k-p}^{k} \mathcal{F}_q^{k+1} \boldsymbol{\pi}_q + \boldsymbol{\eta}''_k \tag{3.5}$$

The three components of the state vector (OD flows, speed–density relationship parameters, capacities) represent distinct aspects of the transportation problem and have different characteristics. Therefore, each of these may evolve over time according to a distinct autoregressive process. This can easily be handled by writing a separate transition equation like the one presented in Equation 3.5 for each such autoregressive process.

3.3 The idea of deviations

Suppose that the model parameters and inputs have been estimated from historical data for several previous days or months. These already estimated (demand and supply) parameters embody a wealth of information about the relationships that affect trip making and traffic dynamics, as well as their temporal and spatial evolution. It is desirable to incorporate as much historical information into the formulation as possible. The most straightforward way

to achieve this is to use *deviations* of the model parameters from best available estimates instead of the actual parameters themselves as state variables. Thus, the model formulation would indirectly take into account all the available *a priori* structural information. The use of deviations has been proposed by Ashok and Ben-Akiva (1993) for the OD estimation and prediction problem.

Using deviations also has other benefits. A normal distribution for the model variables is a useful property for the available statistical tools such as the Kalman Filter extensions used in this research. Traffic flow variables, however, have skewed distributions (unlike the normal distribution which is symmetric). On the other hand, the corresponding deviations of these variables from available estimates would have symmetric deviations and hence are more amenable to approximation by a normal distribution.

The state vector can therefore be expressed as deviations from best historical values: $\Delta\boldsymbol{\pi}_k = \boldsymbol{\pi}_k - \boldsymbol{\pi}_k^H$. The transition equation can easily be reformulated with respect to the new state vector as:

$$\begin{aligned} \boldsymbol{\pi}_{k+1} - \boldsymbol{\pi}_{k+1}^H &= \sum_{q=k-p}^{k} \mathcal{F}_q^{k+1}\left(\boldsymbol{\pi}_q - \boldsymbol{\pi}_q^H\right) + \boldsymbol{\eta}_k \Rightarrow \\ \Delta\boldsymbol{\pi}_{k+1} &= \sum_{q=k-p}^{k} \mathcal{F}_q^{k+1} \cdot \Delta\boldsymbol{\pi}_q + \boldsymbol{\eta}_k \end{aligned} \tag{3.6}$$

Similarly, the direct measurement equation can be written in deviations' form as:

$$\begin{aligned} \boldsymbol{\pi}_k^a - \boldsymbol{\pi}_k^H &= \boldsymbol{\pi}_k - \boldsymbol{\pi}_k^H + \mathbf{v}_k \Rightarrow \\ \Delta\boldsymbol{\pi}_k^a &= \Delta\boldsymbol{\pi}_k + \mathbf{v}_k \end{aligned} \tag{3.7}$$

It should be noted that $\boldsymbol{\pi}_k^a$ and $\boldsymbol{\pi}_k^H$ capture essentially the same thing: an available estimate of the state vector. However, there are subtle differences and —in the interest of generality— a distinction is made. For example, the *a priori* parameters $\boldsymbol{\pi}_k^a$ may correspond to the parameters obtained from the off–line calibration, while the *historical* parameters $\boldsymbol{\pi}_k^H$ may refer to the latest available estimates (e.g. values obtained from the same interval the

previous day).

Finally, the indirect measurement equation can be written as:

$$\begin{aligned} \mathbf{M}_h - \mathbf{M}_h^H &= \mathcal{S}(\boldsymbol{\pi}_h) - \mathbf{M}_h^H + \boldsymbol{\upsilon}_h \Rightarrow \\ \Delta\mathbf{M}_h &= \mathcal{S}(\boldsymbol{\pi}_h^H + \Delta\boldsymbol{\pi}_h) - \mathbf{M}_h^H + \boldsymbol{\upsilon}_h \end{aligned} \tag{3.8}$$

3.4 The model at a glance

The on–line calibration algorithm has been expressed in deviations' form (where Equation 3.6 is the transition equation and Equations 3.7 and 3.8 are the measurement equations). The complete state–space model is shown below for clarity:

$$\begin{aligned} \Delta\boldsymbol{\pi}_{h+1} &= \sum_{q=h-p}^{h} \mathcal{F}_q^{h+1} \cdot \Delta\boldsymbol{\pi}_q + \boldsymbol{\eta}_h \\ \Delta\boldsymbol{\pi}_h^a &= \Delta\boldsymbol{\pi}_h + \mathbf{v}_h \\ \Delta\mathbf{M}_h &= \mathcal{S}(\boldsymbol{\pi}_h^H + \Delta\boldsymbol{\pi}_h) - \mathbf{M}_h^H + \boldsymbol{\upsilon}_h \end{aligned} \tag{3.9}$$

Before moving to the presentation of applicable solution approaches (Chapter 4), it is useful to express the model in the following form:

$$\mathbf{x}_{h+1} = \mathbf{f}(\mathbf{x}_h) + \mathbf{w}_h \tag{3.10}$$

$$\mathbf{y}_h = \mathbf{h}(\mathbf{x}_h) + \mathbf{u}_h \tag{3.11}$$

where Equation 3.10 is the transition equation and Equation 3.11 is the measurement equation.

This form is obtained directly from Equations 3.9 if we denote

$$\mathbf{x}_h = \Delta\boldsymbol{\pi}_h$$

$$\mathbf{y}_t = \begin{bmatrix} \Delta\boldsymbol{\pi}_t^a \\ \Delta\mathbf{M}_t \end{bmatrix}$$

$$\mathbf{f}(\mathbf{x}_t) = \sum_{q=t-p}^{t} \mathcal{F}_q^{t+1}\mathbf{x}_q$$

$$\mathbf{h}(\mathbf{x}_t) = \begin{bmatrix} \Delta\boldsymbol{\pi}_t \\ \mathcal{S}(\boldsymbol{\pi}_t^H + \Delta\boldsymbol{\pi}_t) - \mathbf{M}_t^H \end{bmatrix}$$

$$\mathbf{u}_t = \begin{bmatrix} \mathbf{v}_t \\ \boldsymbol{\upsilon}_t \end{bmatrix}$$

Furthermore, the following assumptions are made on the error vectors $\mathbf{w}_t$ and $\mathbf{u}_t$:

1. $E[\mathbf{w}_t] = 0$

2. $E[\mathbf{w}_t\ \mathbf{w}'_m] = \mathbf{Q}_t\ \delta_{tm}$ where δ_{tm} is the Kronecker delta, i.e. $\delta_{tm} = 1$ if $t = m$ and 0 otherwise $\forall t, m$, and $\mathbf{Q}_t$ is a variance–covariance matrix.

3. $E[\mathbf{u}_t] = 0$

4. $E[\mathbf{u}_t\ \mathbf{u}'_m] = \mathbf{R}_t\ \delta_{tm}$ where δ_{tm} is the Kronecker delta, and $\mathbf{R}_t$ is a variance–covariance matrix.

5. $E[\mathbf{u}_t\ \mathbf{w}'_m] = 0 \quad \forall t, m$, i.e. the errors of the transition and measurement equations are uncorrelated.

These assumptions allow for the derivation of the Kalman Filter–based solution approaches. The assumption of no serial correlation for the transition equation can be defended because the unobserved factors that could be correlated over time are captured by the historical matrix $\boldsymbol{\pi}_t^a$. In some situations (e.g. incidents), however, this assumption might break down. A violation of this assumption, however, can be easily taken care of by using a variant of the estimation algorithm that is described in the following chapter. (An algorithm to handle correlated errors in the transition or measurement equations can be found, for example, in Chui and Chen (1999)).

The assumption of no serial correlation for the measurement equation can be defended using a similar argument. However, this assumption might also break down if, for example,

a specific detector consistently under–estimates or over–estimates a link volume on a particular day. Again, it is easy to relax this assumption and use a variant of the estimation algorithm.

3.5 Conclusion

The on–line calibration approach has been stated as a compact state–space model, comprising transition and measurement equations. While the transition equation 3.10 is linear, the measurement equation 3.11 does not have an analytical expression. This property of the model prohibits the use of the usual Kalman Filtering approaches that have been developed for linear state–space models. However, modified Kalman Filter methodologies have been developed for non–linear models, and can be used instead. Applicable solution approaches are presented in Chapter 4.

Chapter 4

Solution approaches

Contents

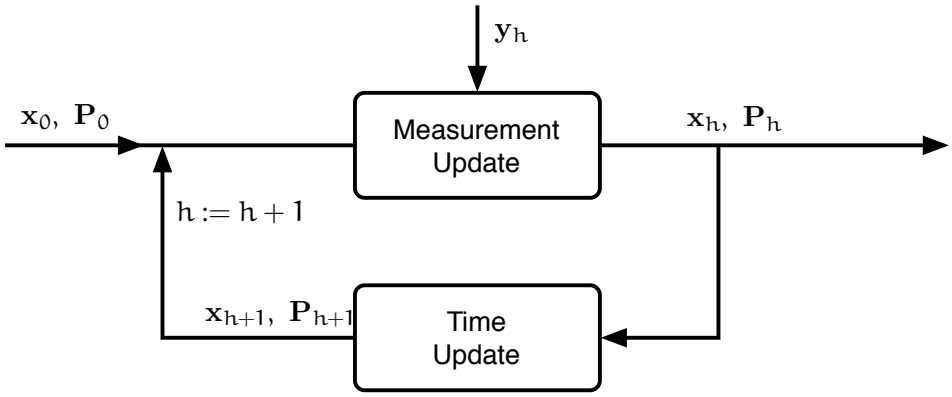

Figure 4-1: Prediction–correction framework of Kalman Filter approach

4.1 Kalman filtering for linear models

The Kalman Filter (Algorithm 4.1) is the optimal minimum mean square error (MMSE) estimator for linear state-space models (Kalman, 1960). While the model formulation of the on–line calibration is not linear (due to the indirect measurement equation), it is still useful to review the basic Kalman Filtering algorithm, since modified Kalman Filter methodologies have been developed for non–linear models.

The Kalman filter provides a recursive solution to the linear optimal filtering problem defined by the following equations

$$\mathbf{X}_{h+1} = \mathbf{F}_h \mathbf{X}_h + \mathbf{w}_h \tag{4.1}$$

$$\mathbf{Y}_h = \mathbf{H}_h \mathbf{X}_h + \mathbf{u}_h \tag{4.2}$$

where $\mathbf{w}_h$ is assumed to be a vector of zero mean, normal and uncorrelated errors with covariance matrix $\mathbf{Q}_h$ and $\mathbf{u}_h$ is assumed to be a vector of zero mean, normal and uncorrelated errors with covariance matrix $\mathbf{R}_h$.

A common way to look at the recursive nature of the Kalman Filter is illustrated in Figure 4-1. This approach is often referred to as "prediction–correction" due to the two steps of time and measurement update.

In words, the main steps of the Kalman Filter are as follows. Suppose that a starting

Algorithm 4.1 Kalman Filter

Initialization

$$\mathbf{X}_{0|0} = \mathbf{X}_0 \tag{4.3}$$
$$\mathbf{P}_{0|0} = \mathbf{P}_0 \tag{4.4}$$

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

Time update

$$\mathbf{X}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{X}_{k-1|k-1} \tag{4.5}$$
$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^{T} + \mathbf{Q}_k \tag{4.6}$$

Measurement update

$$\mathbf{G}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^{T}\left(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^{T} + \mathbf{R}_k\right)^{-1} \tag{4.7}$$
$$\mathbf{X}_{k|k} = \mathbf{X}_{k|k-1} + \mathbf{G}_k\left(\mathbf{Y}_k - \mathbf{H}_k\mathbf{X}_{k|k-1}\right) \tag{4.8}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{G}_k\mathbf{H}_k\mathbf{P}_{k|k-1} \tag{4.9}$$

end for

estimate of the state $\mathbf{X}_0$ is available (Equation 4.3), along with an estimate of the initial state variance–covariance matrix $\mathbf{P}_0$ (Equation 4.4). A *time update* phase makes a prediction of the state (Equation 4.5) and its covariance matrix (Equation 4.6) for the next time interval.

The measurement update phase incorporates the new information about the measurement vector $\mathbf{Y}_k$ and uses it to *correct* the prediction of the state made during the time update. Instrumental in this process is the *Kalman gain* $\mathbf{G}_k$, which is computed as per Equation 4.7. The state can then be updated (*corrected*) using Equation 4.8. Similarly, the state covariance is updated using Equation 4.9.

Further information on the Kalman Filter can be found in many texts, including for example Gelb (1974), Sorenson (1985), and Chui and Chen (1999).

4.2 Towards non–linearity

The original Kalman filter theory applies to linear systems. However, the on–line calibration approach is non–linear (due to the indirect measurement equation). Since many other interesting problems are non–linear, solutions for non–linear models have been sought, leading to the development of modified Kalman Filter methodologies. The most straight-

forward extension is the **Extended Kalman Filter (EKF)**, in which optimal quantities are approximated via first order Taylor series expansion (linearization) of the appropriate equations. The **Iterated EKF** (IEKF) method attempts to improve upon EKF, by using the current estimate of the state vector to linearize the measurement equation in an iterative mode. The different methods have different performance characteristics in terms of computational effort and accuracy of the results (Kalman, 1960; Gelb, 1974).

(I)EKF approaches involve the linearization of the nonlinear measurement and transition functions, around the best available estimate. In the on–line calibration approach, the transition equation is linear, and therefore no linearization is needed. However, differentiation of the measurement equations is required. Since the indirect measurement equation does not —in general— have an analytical expression, analytical derivation is not possible and numerical methods are needed.

The **Unscented Kalman Filter (UKF)** (Julier et al., 1995; Julier and Uhlmann, 1997; Wan et al., 2000; Wan and van der Merwe, 2000; van der Merwe et al., 2000) is an alternative filter. The main difference between the EKF and UKF lies in the representation of the (Gaussian) random variables for propagation through the system dynamics. In the EKF the state distribution is approximated by a random variable which is then analytically propagated through the first order linearization of the non–linear system. For highly non-linear functions this approximation may be very inaccurate. The UKF, on the other hand, uses a deterministic sampling approach (Unscented Transformation, UT) to overcome this issue. The state distribution is again approximated by a random variable and is represented using a (small) number of deterministically selected sample points (often called *sigma points*). These points capture the true mean and covariance of the random variable and, when propagated through the *true* nonlinear system, capture the posterior mean and covariance accurately to the second order (Taylor series expansion) for any nonlinearity (while, as mentioned above, the EKF only reflects the first–order term).

Although the UT requires several approximations, the method tends to be more accurate than many of the other techniques that have been used to propagate random variables through nonlinear transformations (Wan and van der Merwe, 2001). Another advantage of the UKF is that it does not require an analytical expression for the transition and measurement equations. This property makes it directly applicable to simulation–based systems.

(Other techniques can also be used, but additional approximations are generally required. For example, numerical derivatives are necessary in order to use the EKF). Furthermore, the computational complexity of the UKF is of the same order as that of the EKF.

In the remainder of this section three algorithms are outlined (EKF, IEKF and UKF) as they apply to the following state–space model:

$$\mathbf{X}_{h+1} = \mathbf{F}_h \mathbf{X}_h + \mathbf{w}_h \tag{4.10}$$

$$\mathbf{Y}_h = \mathbf{h}(\mathbf{X}_h) + \mathbf{u}_h \tag{4.11}$$

where $\mathbf{w}_h$ is assumed to be a vector of zero mean, normal and uncorrelated errors with covariance matrix $\mathbf{Q}_h$ and $\mathbf{u}_h$ is assumed to be a vector of zero mean, normal and uncorrelated errors with covariance matrix $\mathbf{R}_h$.

The transition equation 4.10 is assumed to be linear (for example, an autoregressive process, such as Equation 3.6). The function $\mathbf{h}(\cdot)$, however, is more general and could describe a simulation–based model. Therefore, the measurement equation 4.11 —in general— does not have an analytical expression (and clearly cannot be assumed to be linear).

4.3 Extended Kalman Filter

The Extended Kalman Filter employs a *linearization* of the non–linear relationship to approximate the measurement equation with a first–order Taylor expansion:

$$\mathbf{H}_h = \left. \frac{\vartheta \mathbf{h}(\mathbf{x}^*)}{\vartheta \mathbf{x}^*} \right|_{\mathbf{x}^* = \mathbf{X}_{h|h-1}} \tag{4.12}$$

In words, the Extended Kalman Filter main steps are as follows. Suppose that a starting estimate of the state $\mathbf{X}_0$ is available (Equation 4.13), along with an estimate of the initial state variance–covariance matrix $\mathbf{P}_0$ (Equation 4.14). A *time update* phase makes a prediction of the state (Equation 4.15) and its covariance matrix (Equation 4.16) for the next time interval.

The measurement update phase incorporates the new information about the measurement

vector $\mathbf{Y}_k$ and uses it to *correct* the prediction of the state made during the time update. The measurement matrix $\mathbf{H}_k$ is obtained through an intermediate linearization step (Equation 4.17). Instrumental in this process is the *Kalman gain* $\mathbf{G}_k$, which is computed as per Equation 4.18. The state can then be updated (*corrected*) using Equation 4.19. Similarly, the state covariance is updated using Equation 4.20.

Further information on the Extended Kalman Filter can be found in many texts, including for example Sorenson (1985), and Chui and Chen (1999).

The on–line calibration approach presented in previous sections does not —in general— have an analytical representation. Therefore, in order to perform the linearization step (Equation 4.17) it is necessary to use *numerical derivatives*. Assuming the use of central derivatives, it is necessary to evaluate the function $2n$ times, where n is the dimension of the state vector. (If forward derivatives are used, then this number drops to $n+1$ evaluations.) Each such evaluation implies one run of the simulator. Therefore, it becomes apparent that this process of linearization dominates the computational complexity of the algorithm.

Algorithm 4.2 Extended Kalman Filter

Initialization

$$\mathbf{X}_{0|0} = \mathbf{X}_0 \tag{4.13}$$
$$\mathbf{P}_{0|0} = \mathbf{P}_0 \tag{4.14}$$

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

Time update

$$\mathbf{X}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{X}_{k-1|k-1} \tag{4.15}$$
$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^{T} + \mathbf{Q}_k \tag{4.16}$$

Linearization

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{h}(\mathbf{x}^*)}{\partial \mathbf{x}^*}\right|_{\mathbf{x}^*=\mathbf{X}_{k|k-1}} \tag{4.17}$$

Measurement update

$$\mathbf{G}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^{T}\left(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^{T} + \mathbf{R}_k\right)^{-1} \tag{4.18}$$
$$\mathbf{X}_{k|k} = \mathbf{X}_{k|k-1} + \mathbf{G}_k\left[\mathbf{Y}_k - \mathbf{h}\left(\mathbf{X}_{k|k-1}\right)\right] \tag{4.19}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{G}_k\mathbf{H}_k\mathbf{P}_{k|k-1} \tag{4.20}$$

end for

4.4 Iterated Extended Kalman Filter

The update step of the EKF involves the linearization of the measurement equation about the present best estimate of the state vector $\mathbf{X}$, i.e., $\mathbf{X}_{h|h-1}$. However, once this step is completed, a presumably superior estimate $\mathbf{X}_{h|h}$ is available which could then be used to linearize the measurement equation and repeat the update step. These iterations could be repeated as many times as deemed necessary. The resulting filter is often called the *Iterated* EKF. Each of these iterations comprise Equations 4.17, 4.18, 4.19, and 4.20.

Note that each iteration of the IEKF involves the linearization step (Equation 4.17). This implies that for each iteration the numerical derivative will need to be re–evaluated. As a consequence, each iteration would increase the overall runtime of the algorithm by an amount equal to the EKF algorithm.

4.5 Unscented Kalman Filter

The **Unscented Kalman Filter (UKF)** (Julier et al., 1995; Julier and Uhlmann, 1997; Wan et al., 2000; Wan and van der Merwe, 2000; van der Merwe et al., 2000) is an alternative filter for dynamic state–space models. The UKF uses a deterministic sampling approach (Unscented Transformation, UT) to represented a random variable using a number of deterministically selected sample points (often called *sigma points*). These points capture the mean and covariance of the random variable and, when propagated through the *true* nonlinear system, capture the posterior mean and covariance accurately to the second order (Taylor series expansion).

4.5.1 The Unscented Transformation

The Unscented Transformation is based on the intuitive expectation that *"with a fixed number of parameters it should be easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function/transformation"* (Julier and Uhlmann, 1996). Following this intuition, one would seek to find a parameterization that would capture the mean and covariance information while at the same time permitting the direct propagation of the information through an arbitrary set of nonlinear equations. This can

be accomplished by generating a discrete distribution having the same first and second (and possibly higher) moments, where each point in the discrete approximation can be directly transformed. The mean and covariance of the transformed ensemble can then be computed as the estimate of the nonlinear transformation of the original distribution.

Given an n–dimensional Gaussian distribution with covariance $\mathbf{P}$, is is possible to generate $O(n)$ points with the same sample covariance from the columns (or rows) of the matrices $\pm\sqrt{n\mathbf{P}}$ (the positive and negative roots). This set of points has a zero mean. However, simply adding the mean $\mathbf{x}$ of the original distribution to each of the points yields a symmetric set of $2n$ points with the desired mean and covariance. Because the set is symmetric its odd central moments are zero, so its first three moments match the original Gaussian distribution.

The main steps of the Unscented Transformation (UT) for calculating the statistics of a random variable that undergoes a nonlinear transformation (e.g. $\mathbf{y_h} = f(\mathbf{x_h})$) are presented in Algorithm 4.3 (Julier and Uhlmann, 1997). Let the n-dimensional random variable $\mathbf{x_h}$ with covariance matrix $\mathbf{P}_{x,h}$ denote the state for time interval h. Since this algorithm also considers the covariance of the measurement vector $\mathbf{P}_{y,h}$ during interval h and the covariance of the state and measurement vectors $\mathbf{P}_{xy,h}$, the covariance of the state vector will be denoted $\mathbf{P}_{x,h} = \mathbf{P}_h$ in order to avoid confusion.

To calculate the statistics of $\mathbf{y}$, a matrix $\mathcal{X}$ is generated using $2n+1$ weighted *sigma points*. $\kappa \in \mathcal{R}$ is a scaling parameter and $\left(\sqrt{(n+\kappa)\mathbf{P}_{x,h}}\right)_i$ is the ith row or column of the matrix square root of $(n+\kappa)\mathbf{P}_{x,h}$. A Cholesky decomposition (Golub and van Loan, 1996) can be used for this step. The value of the scaling parameter κ has a direct effect on the scaling of the points and is an input to the algorithm. The constant a determines the spread of the sigma points around $\bar{\mathbf{x}}$ and is usually set to $0.0001 \leq a \leq 1$. b is used to incorporate prior knowledge of the distribution of $\mathbf{x}$ (for Gaussian distributions, $b = 2$ is optimal). The weights are not time–dependent and do not need to be recomputed for every time interval.

4.5.2 The algorithm

The main steps of the UKF are presented in Algorithm 4.4 (van der Merwe et al., 2000). The initialization step uses the Unscented Transformation (Algorithm 4.3) to generate the $2n+1$ *sigma points* and appropriate weights for the mean and covariance computations. A

Algorithm 4.3 Unscented Transformation

Generation of *sigma points*

$$\mathcal{X}_{0,h} = \mathbf{x_h} \tag{4.21}$$

for $i = 1$ to n **do**

$$\mathcal{X}_{i,h} = \mathbf{x_h} + \left(\sqrt{(n+\kappa)\mathbf{P}_{x,h}}\right)_i \tag{4.22}$$

end for

for $i = n+1$ to $2n$ **do**

$$\mathcal{X}_{i,h} = \mathbf{x_h} - \left(\sqrt{(n+\kappa)\mathbf{P}_{x,h}}\right)_i \tag{4.23}$$

end for

Generation of weights

$$\mathcal{W}_0^m = \kappa/(n+\kappa) \tag{4.24}$$

$$\mathcal{W}_0^c = \kappa/(n+\kappa) + (1-a^2+b) \tag{4.25}$$

for $i = 1$ to $2n$ **do**

$$\mathcal{W}_i^m = \mathcal{W}_i^c = 1/[2(n+\kappa)] \tag{4.26}$$

end for

time and measurement update step is repeated for each run of the algorithm.

The first step in the time update phase is the propagation of the *sigma points* through the transition equation (Equation 4.27). The prior estimate of the state vector is computed as a weighted sum of the propagated sigma points (Equation 4.28). A similar approach is used for the prior estimate of the state covariance (Equation 4.29). The true measurement equation is used to transform the sigma points into a vector of respective measurements (Equation 4.30). The measurement vector is computed as a weighted sum of the generated measurements (Equation 4.31).

The computation of the Kalman gain (and consequently the "correction" phase of the filtering) is based on the covariance of the measurement vector (Equation 4.32) and the covariance of the state and measurement vectors (Equation 4.33). These are computed using the weights (that were obtained from the Unscented Transformation during the initialization step) and the deviations of the sigma points from their means.

The Kalman gain is then computed from these covariance matrices (Equation 4.34). Equation 4.35 introduces the measurement vector $\mathbf{y}_h$ and uses the Kalman gain to correct the

Algorithm 4.4 Unscented Kalman Filter

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

Generate *sigma points* and weights using the Unscented Transformation (Algorithm 4.3)

Time update

$$\mathcal{X}_{k|k-1} = \mathbf{f}(\mathcal{X}_{k-1}) \tag{4.27}$$

$$\mathbf{x}_{k|k-1} = \sum_{i=0}^{2n} W_i^m \mathcal{X}_{i,k|k-1} \tag{4.28}$$

$$\mathbf{P}_{x,k|k-1} = \sum_{i=0}^{2n} W_i^c \left(\mathcal{X}_{i,k|k-1} - \mathbf{x}_{k|k-1}\right) \times \times \left(\mathcal{X}_{i,k|k-1} - \mathbf{x}_{k|k-1}\right)^T + \mathbf{Q}_k \tag{4.29}$$

$$\mathcal{Y}_{i,k|k-1} = \mathbf{h}\left(\mathcal{X}_{i,k|k-1}\right) \tag{4.30}$$

$$\mathbf{y}_{k|k-1} = \sum_{i=0}^{2n} W_i^m \mathcal{Y}_{i,k|k-1} \tag{4.31}$$

Measurement update

$$\mathbf{P}_{y,k} = \sum_{i=0}^{2n} W_i^c (\mathcal{Y}_{i,k|k-1} - \mathbf{y}_{k|k-1}) \times \times (\mathcal{Y}_{i,k|k-1} - \mathbf{y}_{k|k-1})^T + \mathbf{R}_k \tag{4.32}$$

$$\mathbf{P}_{xy,k} = \sum_{i=0}^{2n} W_i^c (\mathcal{X}_{i,k|k-1} - \mathbf{x}_{k|k-1}) \times \times (\mathcal{Y}_{i,k|k-1} - \mathbf{y}_{k|k-1})^T \tag{4.33}$$

$$\mathbf{G}_k = \mathbf{P}_{xy,k}\mathbf{P}_{y,k}^{-1} \tag{4.34}$$

$$\mathbf{x}_k = \mathbf{x}_{k|k-1} + \mathbf{G}_k\left(\mathbf{y}_k - \mathbf{y}_{k|k-1}\right) \tag{4.35}$$

$$\mathbf{P}_{x,k} = \mathbf{P}_{x,k|k-1} - \mathbf{G}_k\mathbf{P}_{y,k}\mathbf{G}_k^T \tag{4.36}$$

end for

state estimate $\mathbf{x}_{\mathsf{h}}$. The state covariance is updated using Equation 4.36.

4.5.3 Correspondence with the Extended Kalman Filter

The EKF and the UKF belong in the same family of algorithms. However, they use different approaches to capture the non–linear transformation of the random variable. The purpose of this section is to demonstrate the correspondence between the two algorithms.

Equation 4.15 captures the propagation of the state using the transition equation. in the EKF. In the UKF the same process is captured through Equations 4.27 and 4.28. The only difference is that instead of using the transition equation on a single state vector, in the UKF all sigma points are propagated through the transition equation and the new prior state estimate is computed as a weighted average.

The covariance propagation is performed in the EKF through Equation 4.16 and in the UKF through Equation 4.29. The second term of both relations is the covariance matrix $\mathbf{Q}_{\mathsf{h}}$. The first term of Equation 4.16 includes the state covariance and the transition matrix $\mathbf{F}$. The first term of Equation 4.29 is a weighted sum of the square of the error vector (and therefore an estimate of the covariance).

The Kalman gain computation in the EKF (Equation 4.18) involves the state covariance in the numerator and a function of the measurement covariance plus the error covariance $\mathbf{R}_{\mathsf{h}}$ at the denominator. In the UKF, the computation of the Kalman gain (Equation 4.34) is a function of the state and measurement covariance on the numerator and two terms in the denominator. One of these terms is the error covariance $\mathbf{R}_{\mathsf{h}}$. The other term is an estimate of the covariance of the measurement vector (obtained as a weighted sum of the squares of the deviations of the measurement).

Finally, the last two steps are almost identical (state estimate correction: Equation 4.19 in the EKF and Equation 4.35 in the UKF, and state covariance correction: Equation 4.20 in the EKF and Equation 4.36 in the UKF).

In conclusion, it should be noted that for the state–space model considered in this section, the full power of the UKF is not exhibited, because the transition equation is linear. If both the transition and measurement equations were non–linear, then the EKF would approximate both through a first–order Taylor expansion (thus introducing another degree of

approximation).

An application of the presented solution approaches follows.

4.6 Application

The three algorithms (EKF, IEKF, and UKF) have been applied to the problem of on–line calibration of speed–density relationship parameters for single sensors. This is a subset of the on–line calibration approach and does not involve capacity or demand parameter estimation, nor are interactions with other sensors considered. The objective of this application is to demonstrate the algorithms and obtain intuition regarding their applicability in the context of the on–line calibration.

Sensor data from freeway I-405 in Irvine, CA, and freeway M-27 in Southampton, U.K. have been used. Morning period (4:00am to 10:00am) data have been used for the sensor from Irvine, CA, since this period includes the peak flow for this sensor. Speed and density data are available in 30–second intervals. The peak flow for the sensor from Southampton, U.K., is observed in the evening. Therefore, afternoon/evening period (12:00noon to 8:00pm) data are used. Speed and density data for this sensor are available in 1–minute intervals.

The following speed–density relationship has been assumed:

$$u = u_f \left[1 - \left(\frac{\max(0, K - K_{min})}{K_{jam}} \right)^{\beta} \right]^{\alpha} \tag{4.37}$$

where u denotes the speed, u_f is the free flow speed, K is the density, K_{min} is the minimum density, K_{jam} is the *jam* density and α and β are model parameters. The parameter vector would then become $\Pi_h = \begin{bmatrix} \alpha_h & \beta_h & K_{min,h} & K_{jam,h} & u_{f,h} \end{bmatrix}^T$.

A priori estimates of the parameter values are obtained by fitting the speed–density relationship to an initial set of data (three days for Irvine and five days for Southampton) using non-linear least squares. Data from a different day are used for the on–line calibration.

The *a priori* estimates of the parameters are used as the initial state for the on–line calibration. A random walk is assumed for the autoregressive process. The error covariance matrix of the measurement vector has been computed from the surveillance data used for the off–line calibration. It is not possible to compute an error covariance matrix for the state

Table 4.1: Summary results.

		Est	improv	1Pred	improv	2Pred	improv
Irvine	Offline	0.0709	–	0.0733	–	0.0755	–
	EKF	0.0595	16.08%	0.0675	7.91%	0.0683	9.54%
	IEKF	0.0547	22.85%	0.0665	9.28%	0.0675	10.60%
	UKF	0.0634	10.58%	0.0656	10.50%	0.0685	9.27%
Southampton	Offline	0.119	–	0.119	–	0.119	–
	EKF	0.058	51.01%	0.107	10.08%	0.109	8.40%
	IEKF	0.0538	54.79%	0.102	14.29%	0.102	14.29%
	UKF	0.0829	30.34%	0.0907	23.78%	0.0965	18.91%

vector, since a single value is available for each parameter (from the off–line calibration). Therefore an *ad-hoc* covariance matrix has been used where the variance of each parameter is proportional to its magnitude.

The performance of each algorithm is assessed using the normalized root mean square error (RMSN) of the speeds:

$$\mathrm{RMSN} = \frac{\sqrt{N \sum_N (u - \hat{u})^2}}{\sum_N u}$$

where N is the number of measurements and $\hat{u}$ denotes estimated (predicted) speeds.

Table 4.1 shows the summary results for the EKF, the IEKF (after two iterations) and the UKF algorithms. Estimated and predicted speeds using the off–line calibrated parameters are used as the base case. The algorithm is executed at a 15–minute interval. Therefore, one–step prediction corresponds to prediction 15 minutes into the future, while two–step prediction reflects conditions 30 minutes into the future. The RMSN values obtained after the speed–density parameters were calibrated on–line are presented, as well as percent improvement for each case (over the reference case).

All on–line calibration algorithms provide significant advantages, both for estimation and (one– and two–step) prediction. The results in general follow the intuitive expectation that accuracy decreases from estimation to prediction. An interesting observation is that while the (I)EKF provides a better fit for estimation, in this application it is usually outperformed by the UKF for prediction (except for 2–step prediction for the Irvine network). This property makes UKF an appealing algorithm, since naturally prediction of performance is more critical in this context.

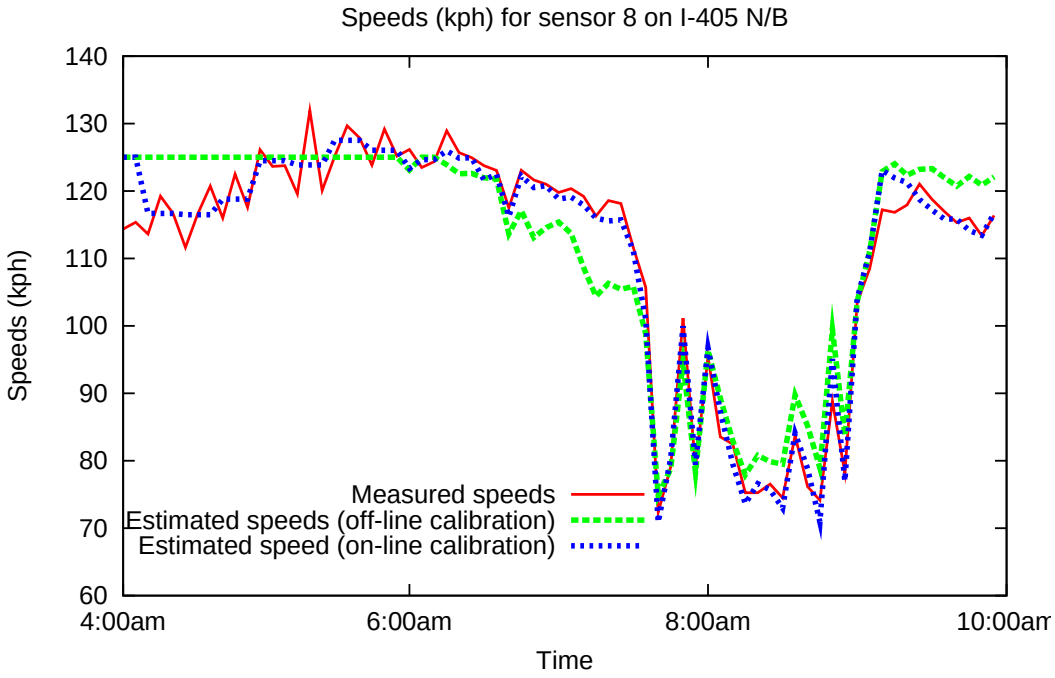

Figure 4-2: Estimated speeds for Irvine, CA. (4:00am to 10:00am)

Figure 4-2 shows the on–line estimated speeds, as well as the measured speeds and the speeds obtained when the off-line calibrated parameters are used for Irvine (for clarity of presentation, speeds are aggregated in 5–minute intervals). One–step and two–step prediction (i.e. 30 minute) results are shown in Figures 4-3 and 4-4 respectively. (Since a 15 minute interval is assumed, one–step prediction means prediction 15 minutes into the future, while two–step prediction means prediction 30 minutes into the future).

It becomes apparent that on-line calibration provides clear benefits for the uncongested regime, where variations in the free–flow speed are tracked more accurately. Furthermore, in the transition from the free–flow regime to the peak period (i.e. from approximately 6:30 to 7:30am), the speeds obtained from the off-line calibration are significantly lower than those measured. On the other hand, speeds calculated using the on–line calibrated parameters are considerably closer to the observed speeds.

Similarly, Figure 4-5 shows the on–line estimation results for Southampton (for visualization purposes, speeds are again aggregated in 5–minute intervals). One–step and two–step prediction (i.e. 30 minute) results are shown in Figures 4-6 and 4-7 respectively. The impact of on–line calibration becomes particularly evident during the recovery of the speed in the

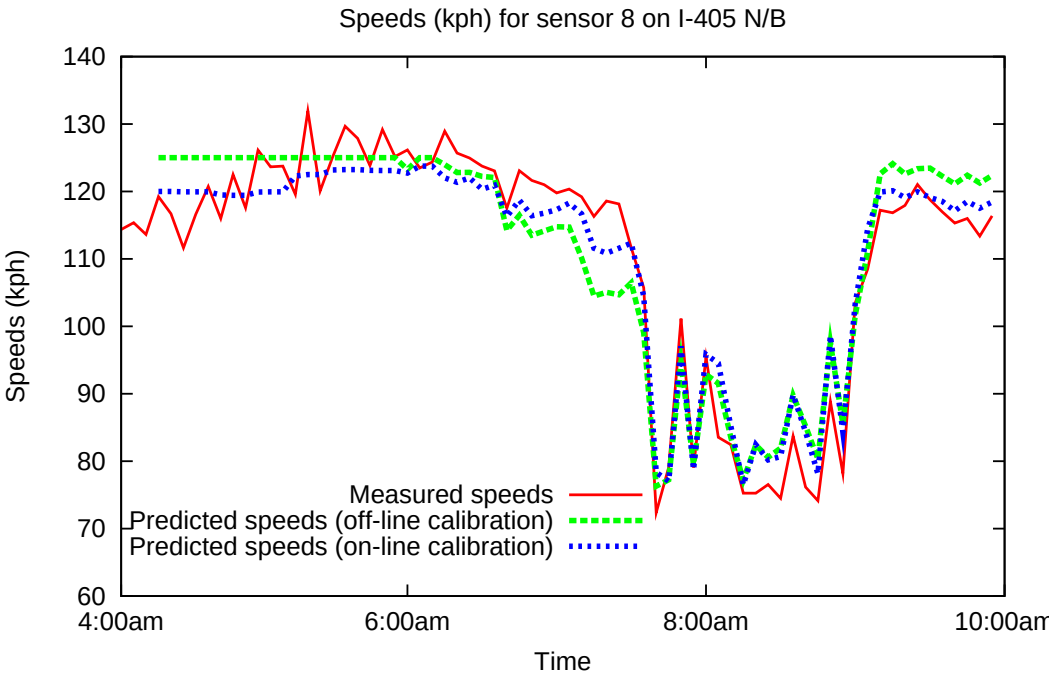

Figure 4-3: One–step predicted speeds for Irvine, CA. (4:00am to 10:00am)

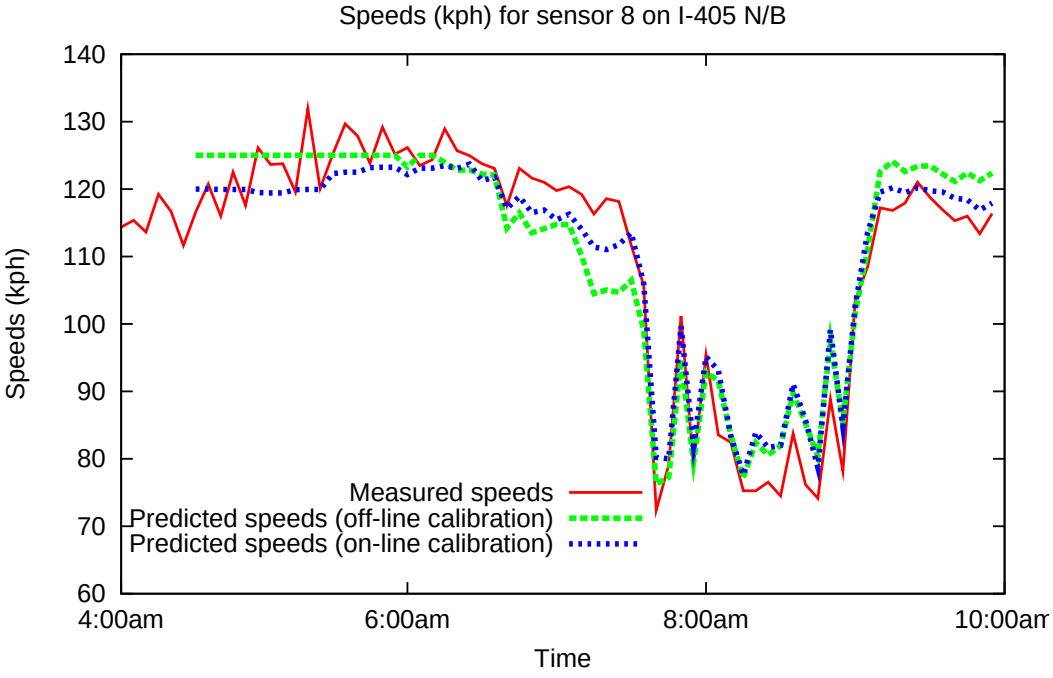

Figure 4-4: Two–step predicted speeds for Irvine, CA. (4:00am to 10:00am)

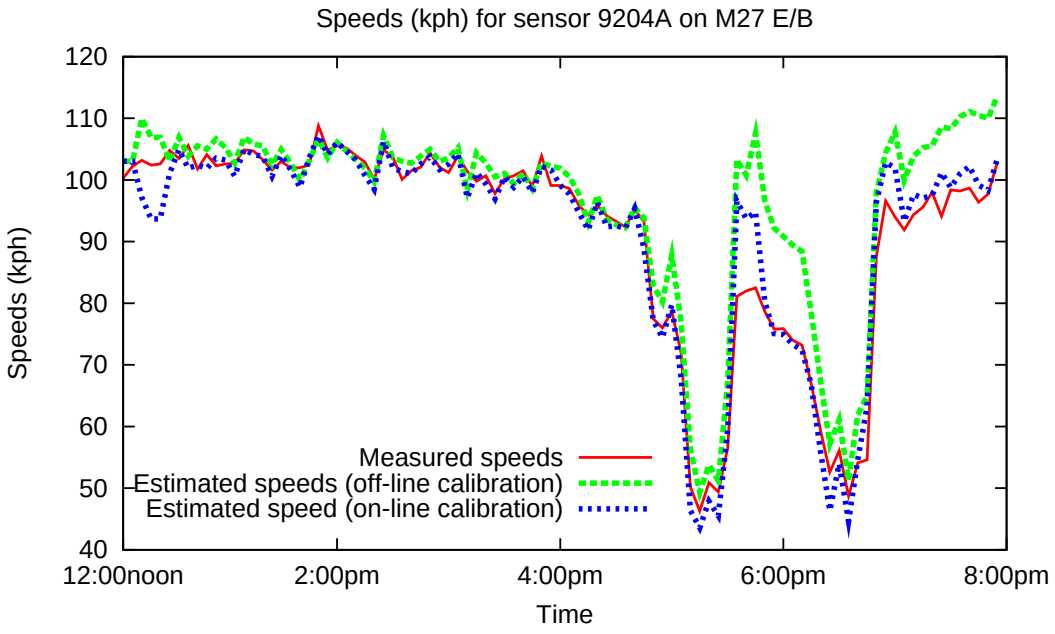

Figure 4-5: Estimated speeds for Southampton, U.K. (12:00noon to 8:00pm)

middle of the peak period (around 6:00pm), as well as following the peak (between 7:00 and 8:00pm).

In order to obtain some further insight into the performance of the algorithms, it is useful to look at the evolution of the estimated parameters over time. Figure 4-8 shows the variability of the parameters for the sensor from the Irvine network. (Jam density and minimum density were not affected by the on–line calibration, but instead stayed close to their original values. Therefore, only the free–flow speed and the two exponents are shown).

Both algorithms capture the same trends, albeit with different magnitude. Free–flow speed estimated by both algorithms is very similar. The main effect is that the free–flow speed is increased before the peak period. Both algorithms capture a small decrease in the value of the α parameter at the beginning of the peak period. The EKF algorithm then switches to a small increase during the peak period. Following the peak period, both algorithms return to the original value.

The reverse trend is captured by the estimated β parameter values. In particular, at the beginning of the peak period both algorithms increase the estimated value (the UKF results in a larger increase), while at the end of the peak period the EKF captures a decrease. Following the end of the peak period both algorithms return to the original value of the

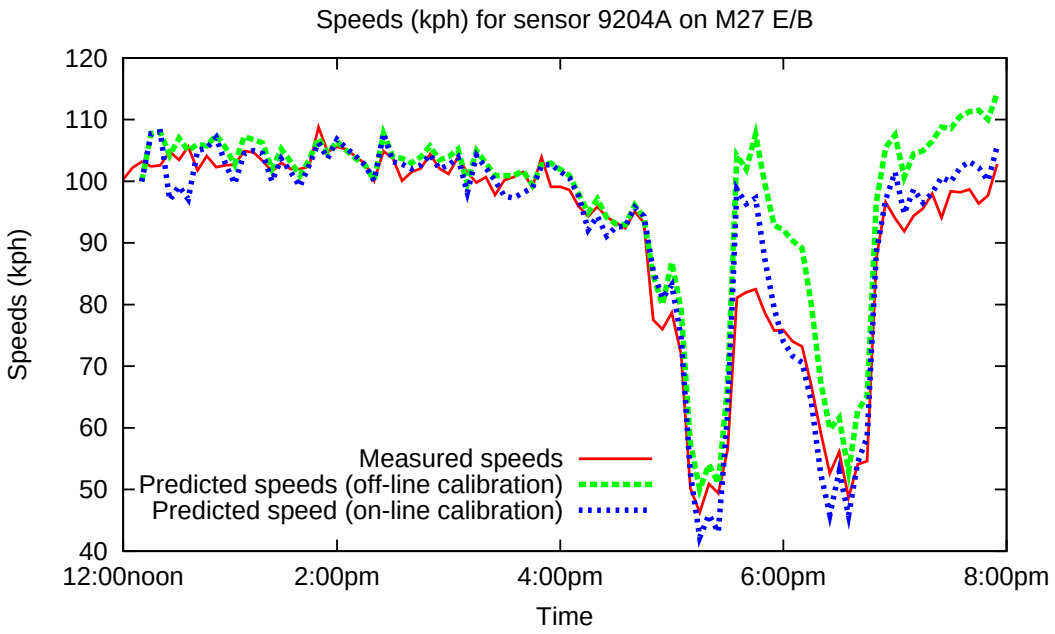

Figure 4-6: One–step predicted speeds for Southampton, U.K. (12:00noon to 8:00pm)

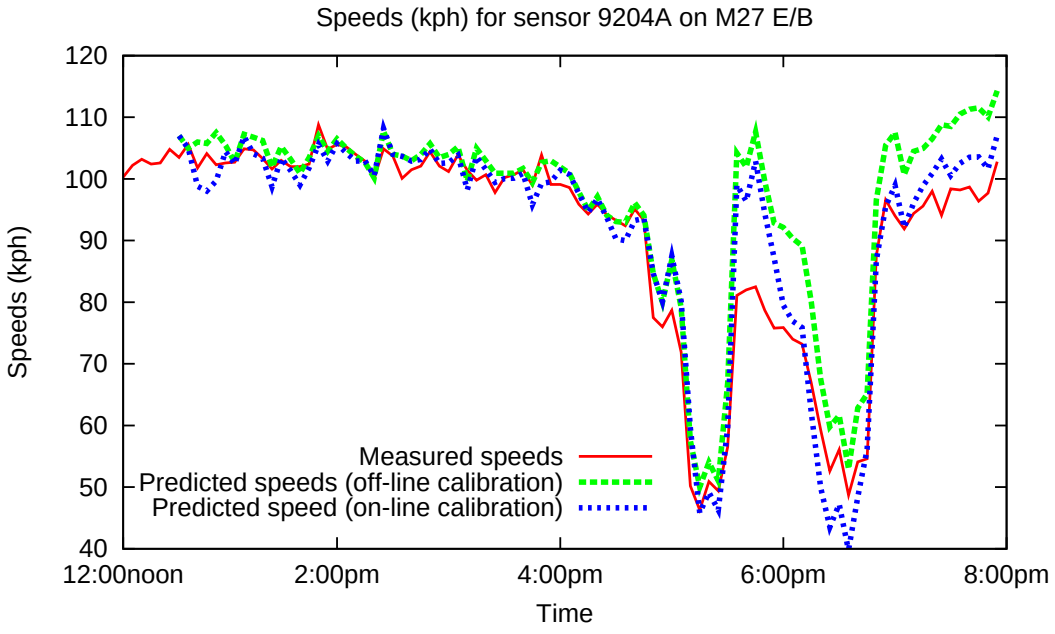

Figure 4-7: Two–step predicted speeds for Southampton, U.K. (12:00noon to 8:00pm)

parameter.

Figure 4-9 shows the variability of the parameters for the sensor from the Southampton network. (Again jam density and minimum density were not affected by the on–line calibration, but instead stayed close to their original values. Therefore, only the free–flow speed and the two exponents are shown).

Again, both algorithms capture similar trends in the evolution of the parameters. The UKF reflects a small decrease in free–flow speed during the peak period (from 130kph to 122kph a little before 6:00pm), while only a marginal decrease is indicated by the EKF.

The reverse trend is observed in the estimated values for the α parameter, with the EKF capturing moderate variations of the parameter (around its original value). The UKF algorithm captures the same trends (especially during the peak period, i.e. between 5:00 and 7:00pm) but at a much smaller scale.

Finally, very similar trends are reflected in the estimated values for the β parameter, with both algorithms capturing a reduction in the value of the parameter during, and after, the peak period. The EKF indicates a slightly larger decrease.

The above analysis of the estimated parameters indicates that both algorithms capture the same underlying trends, but assign different weights to each parameter. A further conclusion is that the minimum and jam density are not changed, but instead the changes are reflected in the free–flow speed and the two parameters (α and β). This finding suggests that it may be meaningful to maintain the values of the minimum and jam densities at their starting values, and only update the remaining three parameters online.

This observation suggests that there may be an issue of observability, due to the relatively small number of available data used to estimate several parameters. In particular 30 data points are used to estimate 5 parameters in the Irvine data, while 15 data points are used to estimate 5 parameters in the Southampton data.

Figure 4-10 summarizes the estimation performance of the Iterated Extended Kalman Filter as a function of the number of iterations for the two networks. The off-line calibration is provided as the reference. The performance of the IEKF solution algorithm stabilized after a few iterations for both networks. The RMSN value obtained when the off–line calibrated parameters are used is 0.0709 for the Irvine network and 0.119 for the Southampton network.

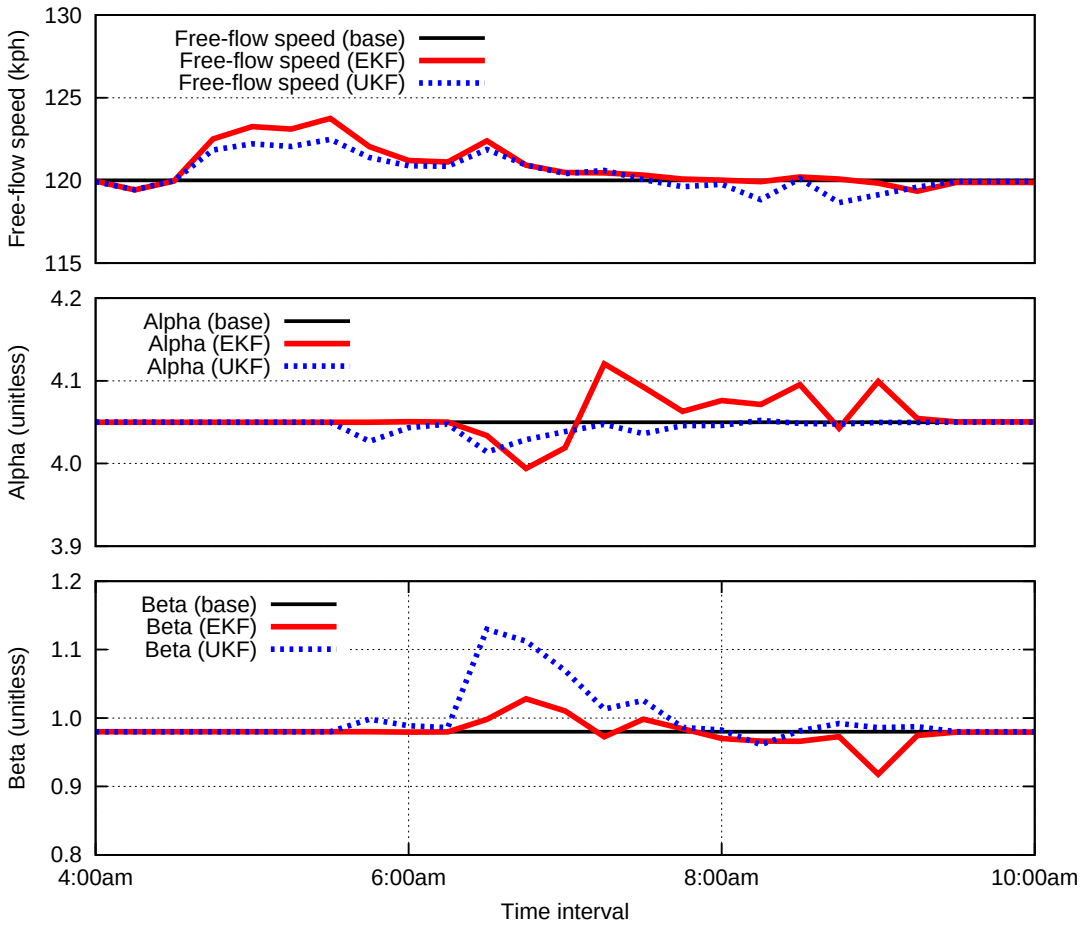

Figure 4-8: Estimated parameters (Irvine, CA)

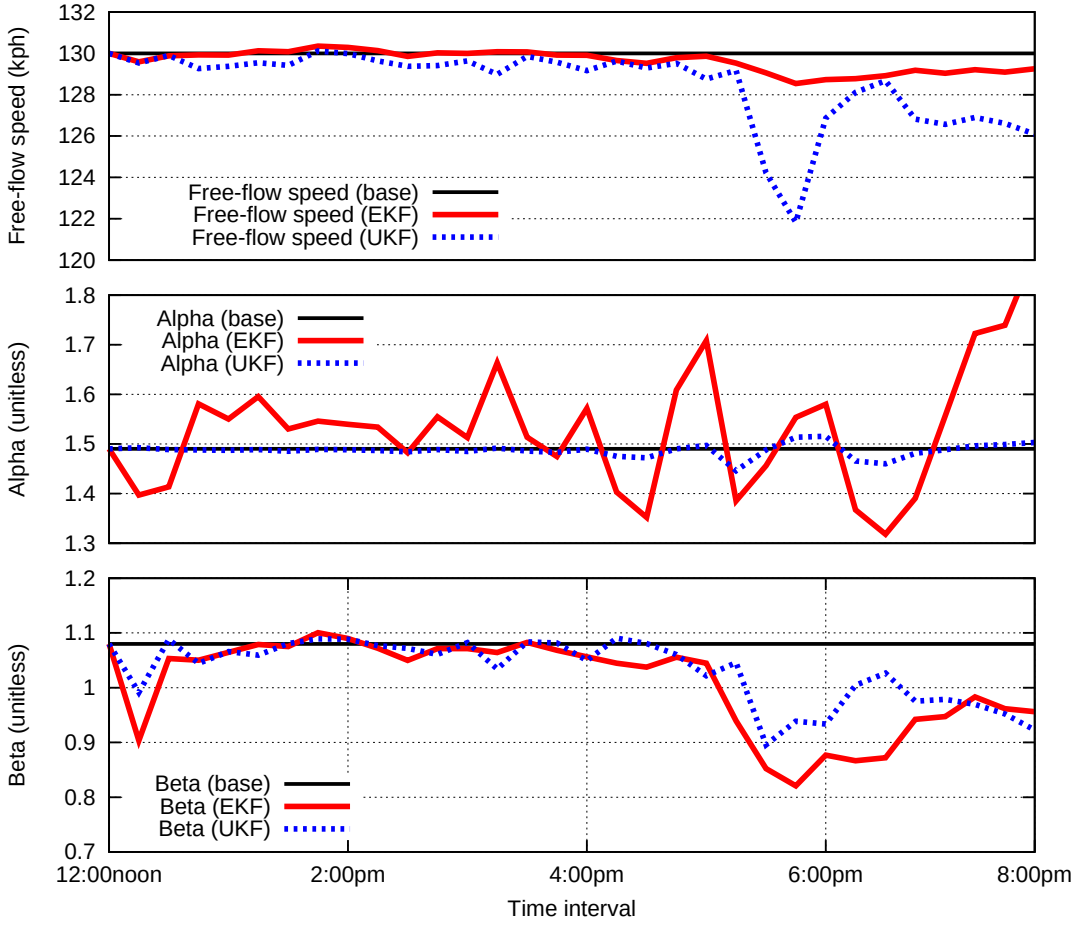

Figure 4-9: Estimated parameters (Southampton, U.K.)

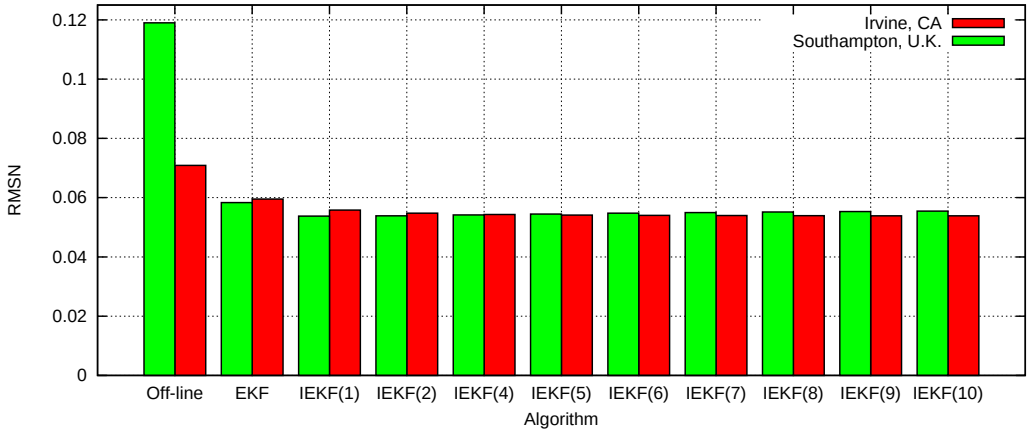

Figure 4-10: Performance of Iterated EKF

Using the parameters estimated from the EKF, the RMSN drops to 0.0595 (an improvement of 16.1%) for Irvine, and to 0.0583 (an improvement of 51%) for Southampton. After only two iterations of the IEKF, the RMSN drops to 0.0547 (an improvement of 22.8% from the reference case and a 8.1% improvement from the EKF) for Irvine and 0.0538 (an improvement of 54.8% from the reference case and a 7.7% improvement from the EKF) for Southampton. As expected, the use of the Iterated EKF is beneficial (in terms of accuracy). It should be noted that each iteration of the IEKF involves roughly the same computational cost as the EKF. In this case study, it appears that a small number of iterations of the IEKF may be sufficient to obtain a good solution.

4.7 Conclusion

Applicable solution approaches for on–line calibration have been discussed and three of them (EKF, IEKF, and UKF) have been presented in detail. The algorithms have been applied to the problem of speed–density relationship calibration, using freeway sensor data. The improvement in the estimation of speeds due to on–line calibration (compared with the speeds obtained from the off–line calibrated relationship) is demonstrated. The EKF provides the most straightforward solution to this problem, and indeed achieves considerable improvements in estimation and prediction accuracy. The additional benefits obtained from a —more computationally expensive— Iterated EKF algorithm are shown.

An innovative solution technique (the UKF) is also presented. The UKF has a number of

unique qualities and advantages over the EKF, including no need for explicit computation of derivatives. Instead, the UKF uses the Unscented Transformation (UT) to compute the necessary statistics. Thus, there is no need to linearize the measurement equation (which leads to an approximation in the (I)EKF models, which is perhaps the biggest criticism of these models).

Chapter 5

Practical Considerations

Contents

In this chapter, practical considerations associated with the on–line calibration approach are presented and approaches to deal with them are described. Data considerations are discussed first.Computational considerations are discussed next and a problem decomposition approach is suggested, followed by a discussion on the potential use of parallelization techniques. This chapter concludes with the presentation of an approximation of the Kalman Filter algorithm with very desirable computational properties.

5.1 Data considerations

This section starts with a discussion of the data requirements of the on–line calibration approach and the possibility that some of the required information will not be available. A staged–approach that addresses this issue is outlined, where the initial "warm–up" phase uses only available data as input. The output of this stage can be used to estimate the remaining inputs for the second phase (which reflects the full on–line calibration approach, as presented in Chapter 3).

5.1.1 Data requirements

The on–line calibration approach assumes the following types of information, which were described in Section 2.1.1:

- Historical traffic information describing the transportation system;
- Surveillance data capturing the prevailing traffic conditions; and
- *A priori* values of the model inputs and parameters.

We now turn our attention to the *a priori* values of the model inputs and parameters, which enter the state–space model through the direct measurement equation (Equation 3.7):

$$\Delta\boldsymbol{\pi}_{\mathrm{h}}^{\mathrm{a}} = \Delta\boldsymbol{\pi}_{\mathrm{h}} + \mathbf{v}_{\mathrm{h}} \tag{5.1}$$

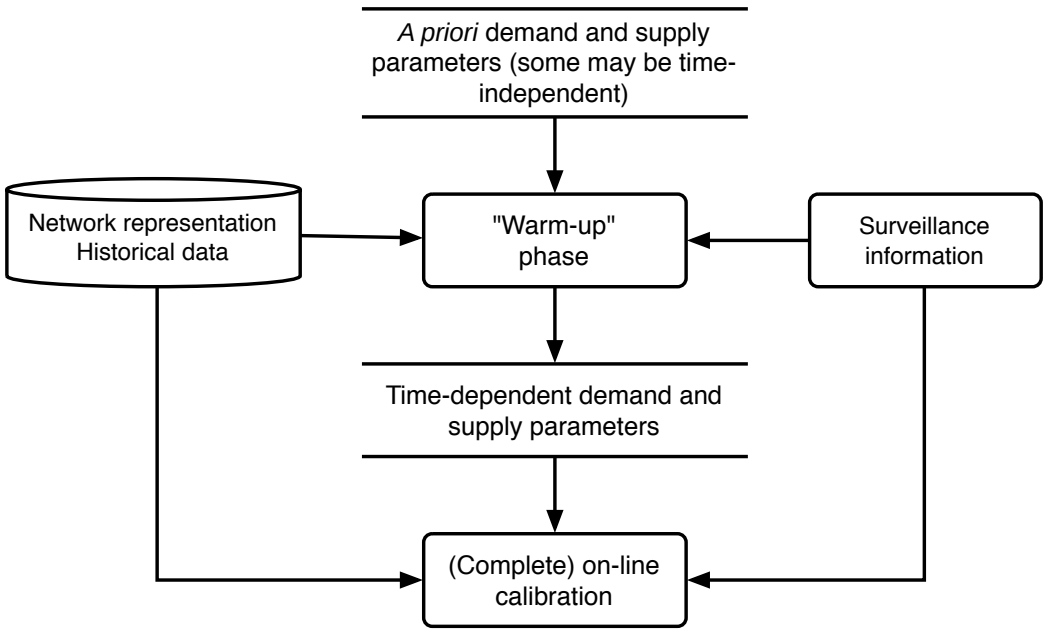

Figure 5-1: "Warm–up" phase

where the variance/covariance matrix of the error vector $\mathbf{v}_h$ is required. Furthermore, the autoregressive factors $\mathcal{F}_q^{h+1}$ for the transition equation (Equation 3.6):

$$\Delta\boldsymbol{\pi}_{h+1} = \sum_{q=h-p}^{h} \mathcal{F}_q^{h+1} \cdot \Delta\boldsymbol{\pi}_q + \boldsymbol{\eta}_h \tag{5.2}$$

need to be estimated.

The estimation of both the covariance matrix of the error vector $\mathbf{v}_h$ and the autoregressive factors $\mathcal{F}_q^{h+1}$ is possible if time–dependent estimates of the state vector are available. It is, however, possible that only mean (i.e. not time–dependent) off–line calibrated values will be available for some parameters. Without loss of generality, for the following discussion it is assumed that time–dependent values are not available for the supply parameters. In that case, relevant autoregressive factors and error covariances cannot be estimated.

In the following section, an approach to overcome this problem and initialize the on–line calibration using limited data is presented.

5.1.2 "Warm–up" phase

A flexible approach that employs a "warm–up " phase to address this issue is presented in Figure 5-1. The assumption of time–independent supply parameter values (and therefore the inability to estimate autoregressive factors) limits the transition equation of the supply parameters to the simplest possible autoregressive process, that of degree one [AR(1)]. (Such autoregressive processes are also often referred to as *random walks*.) (Greene, 2000) This process would output time–dependent demand and supply parameters. Thus, all required inputs for the second phase —the application of the full approach— would be available.

The transition equation (Equation 3.6 or 5.2) can be replaced by the following two equations:

$$\Delta \boldsymbol{x}_{h+1} = \sum_{q=h-p}^{h} \mathcal{F}_q^{h+1} \cdot \Delta \boldsymbol{x}_q + \boldsymbol{\eta}_h^x \tag{5.3}$$

$$\Delta \boldsymbol{\gamma}_{h+1} = \Delta \boldsymbol{\gamma}_h + \boldsymbol{\eta}_h^\gamma \tag{5.4}$$

where $\Delta \boldsymbol{x}_h$ is a vector of deviations of OD flows departing during interval h from available estimates, $\Delta \boldsymbol{\gamma}_h$ is a vector of deviations of the supply parameters from available estimates, $\boldsymbol{\eta}_h^x$ is a vector of error terms associated with the OD flows, and $\boldsymbol{\eta}_h^\gamma$ is a vector of error terms associated with the supply parameters.

Due to the modification of the second part of the transition equation (Equation 5.4), this reformulation does not require autoregressive factors for the supply parameters. However, the covariance matrix of the vector of error terms $\boldsymbol{\eta}_h^\gamma$ (associated with the supply parameters) is also unknown, since only estimated mean values of these parameters are available at this stage. Initial, *ad hoc* values of this covariance matrix need to be assumed.

Furthermore, the direct measurement equation (Equation 3.7 or 5.1) can be rewritten as follows:

$$\Delta \boldsymbol{x}_h^a = \Delta \boldsymbol{x}_h + \mathbf{v}_h^x \tag{5.5}$$

$$\Delta \boldsymbol{\gamma}_h^a = \Delta \boldsymbol{\gamma}_h + \mathbf{v}_h^\gamma \tag{5.6}$$

Since only static estimates of the supply parameters have been assumed, the covariance matrix of the vector $\mathbf{v}_h^\gamma$ is unknown. Initial, *ad hoc* values of this covariance matrix also need to be assumed.

The complete modified state–space model for the "warm–up" phase would comprise the following equations:

$$\begin{aligned}
\Delta\mathbf{x}_{h+1} &= \sum_{q=h-p}^{h} \mathcal{F}_q^{h+1} \cdot \Delta\mathbf{x}_q + \boldsymbol{\eta}_h^x \\
\Delta\boldsymbol{\gamma}_{h+1} &= \Delta\boldsymbol{\gamma}_h + \boldsymbol{\eta}_h^\gamma \\
\Delta\mathbf{x}_h^a &= \Delta\mathbf{x}_h + \mathbf{v}_h^x \\
\Delta\boldsymbol{\gamma}_h^a &= \Delta\boldsymbol{\gamma}_h + \mathbf{v}_h^\gamma \\
\Delta\mathbf{M}_h &= \mathcal{S}\left(\boldsymbol{\pi}_h^H + \Delta\boldsymbol{\pi}_h\right) - \mathbf{M}_h^H + \boldsymbol{\upsilon}_h
\end{aligned} \tag{5.7}$$

The output of this intermediate —or "warm–up" phase— is time–dependent demand and supply parameters. The only remaining step for the application of the full model is the autoregressive factors and the covariance matrices for the error vectors $\boldsymbol{\eta}_h^\gamma$ and $\mathbf{v}_h^\gamma$. Practical approaches to estimate these missing autoregressive factors and error covariances are presented next.

5.1.3 Estimation of autoregressive factors

Let the matrix $\mathcal{F}_q^{\gamma,h+1}$ represent the effect of deviations in the supply parameters γ in time interval q on the respective deviations in interval $h+1$. Let us further denote the elements of this matrix by $\mathbf{f}_{r',q}^{r,h+1}$, which captures the effect of the deviation in the r'th supply parameter during interval q on the deviation in the rth parameter during interval $h+1$. Estimation of the matrix could be done element by element for each interval. The factor $\mathbf{f}_{r',q}^{r,h+1}$ could be estimated through a regression of the form:

$$\Delta\boldsymbol{\gamma}_{r,h+1} = \sum_{q=h+1-p'}^{h} \left(\mathbf{f}_{1,q}^{r,h+1}\Delta\boldsymbol{\gamma}_{1,q} + \cdots + \mathbf{f}_{n_\gamma,q}^{r,h+1}\Delta\boldsymbol{\gamma}_{n_\gamma,q}\right) + \omega'_{r,h+1} \tag{5.8}$$

where n_γ is the number of supply parameters, p' is the degree of the autoregressive process and represents the number of lagged intervals whose supply parameter deviations contribute

to the respective deviations for interval $h+1$, and $\omega'_{r,h+1}$ is an error term. n_γ such regressions would be needed in order to obtain the entire matrix $\mathcal{F}_q^{\gamma,h+1}$. Moreover, one would have to obtain such a matrix for each time interval h. In other words, each day of estimated supply parameters would yield exactly one observation. As a consequence, a large data set encompassing sufficient days of application of the "warm–up" phase would be required.

In the absence of such a large data set, simplifications to the structure of the autoregressive matrices can make the estimation of the factors feasible. For example, it might be reasonable to assume that the structure of the autocorrelation does not vary with respect to time interval h. This would imply that the elements of $\mathcal{F}_q^{\gamma,h+1}$ would depend only on the difference $h-q$, and not on the individual values of h and q. It is now possible to write equations such as Equation 5.8 for each interval within one day and have enough observations to estimate the elements of the matrix even with data from a single day.

The problem can be further simplified by making the reasonable assumption that deviations in the value of the rth supply parameter would be affected primarily by the preceding deviations of the same parameter, and that contributions from other parameters would be insignificant in comparison. The resulting regression would assume a much simpler form:

$$\Delta\boldsymbol{\gamma}_{r,h+1} = \sum_{q=h+1-p'}^{h} \mathbf{f}_{r,q}^{r,h+1}\Delta\boldsymbol{\gamma}_{r,q} + \omega_{r,h+1} \tag{5.9}$$

and the resulting $\mathcal{F}_q^{\gamma,h+1}$ matrix would be diagonal (with element $\mathbf{f}_{r,q}^{r,h+1}$ being the rth element in the diagonal). The value of p', representing the number of lagged intervals whose supply parameter deviations contribute to the respective deviations for interval $h+1$, would be obtained from statistical significance test on regression coefficients for various lags.

5.1.4 Estimation of error covariances

The missing error covariances can be obtained in a fairly straightforward way. The matrix $\mathbf{Q}_h^\gamma$ would be obtained by OLS regressions on Equations 5.8. The (i,j)th element of this

matrix could be approximated by

$$\mathbf{Q}^{\gamma}_{i,j,h} = \mathbf{e}'_{i,h}\mathbf{e}_{j,h}/n \tag{5.10}$$

where $\mathbf{e}$ is the OLS residual vector and n is the number of sample observations. The above equation assumes dependence on time interval h. This can be relaxed by assuming that the structure of the autocorrelation remains constant. In that case the matrix $\mathbf{Q}^{\gamma}_{i,j,h}$ would reduce to $\mathbf{Q}^{\gamma}_{i,j}$.

Similarly one can obtain the covariance matrix $\mathbf{R}^{\gamma}_{h}$ from the residuals $\mathbf{d}_h$ of the indirect measurement equation 5.6. These residuals would be obtained from computing the differences:

$$\mathbf{d}_h = \Delta\boldsymbol{\gamma}^{a}_{h} - \Delta\boldsymbol{\gamma}_h \tag{5.11}$$

Each day would yield one value for every residual vector $\mathbf{d}_h$. The covariance matrices $\mathbf{R}^{\gamma}_{h}$ can be calculated from the values of these residual vectors over several days. The process can be simplified by assuming that the covariance matrix $\mathbf{R}^{\gamma}_{h}$ is time–invariant. Another reasonable approach is to stratify the time periods in a small number of groups, during which the covariance matrix would be invariant, and combine the observations from each group for the estimation of a single matrix.

This methodology could be extended to the demand inputs as well, if needed. In particular, if only mean estimated OD flows are available, then the transition equation for the OD flows can also be modified. Application of the "warm–up" phase would then provide time–dependent OD flows, from which autoregressive factors and covariance matrices can be computed.

Given the output of this "warm–up" phase, it is then possible to perform the "complete" on–line calibration approach, as defined in Section 3.4.

5.2 Computational considerations

Computational considerations associated with the solution of the on–line calibration approach are addressed in this section. A discussion on the problem dimension (Section 5.2.1)

helps put the issue into perspective. Three different approaches to improve on the runtime performance are presented. Problem decomposition approaches that reduce the size of the problem to be solved using the computationally intensive, simulation–based algorithms presented in Chapter 4 are discussed in Section 5.2.2. Parallelization approaches that can reduce the overall runtime by distributing the computational load to additional resources are discussed in Section 5.2.3. Finally, an approximation algorithm that completely eliminates the need for on–line function evaluations (by relying instead on information obtained off–line) is presented in Section 5.2.4.

5.2.1 Problem dimension

The on–line calibration approach estimates the demand and supply parameters jointly, thus considering their interactions and attributing the effects to both components. The computational cost of solving this problem is exacerbated by the fact that there is no analytical formulation for the indirect measurement equation (Equation 3.8):

$$\Delta \mathbf{M}_{h} = \mathcal{S}\left(\boldsymbol{\pi}_{h}^{H} + \Delta\boldsymbol{\pi}_{h}\right) - \mathbf{M}_{h}^{H} + \boldsymbol{\upsilon}_{h} \tag{5.12}$$

While approaches to solve such problems do exist (see Chapter 4), they require a large number of function evaluations. In the case of the Extended Kalman Filter (EKF) — and given the lack of an analytical function for the indirect measurement equation— it is necessary to use numerical derivatives. If central numerical derivatives are used, the number of function evaluations is $2n$, where n is the dimension of the state vector. Similarly, the Unscented Kalman Filter requires $2n + 1$ function evaluations. In the case of a simulation–based DTA system, each evaluation implies one run of the simulator, which can be computationally very expensive.

A closer look at the components of the state vector can be helpful in putting the problem dimension into perspective. As mentioned previously, the state vector comprises OD flows, speed–density relationship parameters and segment capacities. The number of OD flows used to capture the traffic demand patterns can be in the hundreds.

On the supply side, a network can be represented using hundreds of segments. The capacities of the segments contribute to the dimension of the state vector. Furthermore, speed–density

relationships require a small number of parameters for each link or segment. For example, if the following relationship is used:

$$u = u_f \left[1 - \left(\frac{\max(0, K - K_{min})}{K_{jam}} \right)^{\beta} \right]^{\alpha} \tag{5.13}$$

where u is a speed measurement, u_f is the free flow speed, K is the density, K_{min} is the minimum density, K_{jam} is the *jam* density and α and β are model parameters, five additional parameters would have to be added to the state vector for each segment.

In the remainder of this chapter we look at three approaches to improving the computational properties of this model. First, a problem decomposition approach is presented, which exploits the available analytical approximation for the OD estimation problem and the speed–density relationship in order to reduce the size of the problem to be solved using computationally intensive function evaluations. A parallelization approach that allows for the reduction of the total runtime of the algorithm (at the cost of additional resources) is presented next. Finally, an approximate algorithm that exploits information that can be pre–computed off–line to eliminate the need for on–line function evaluations is presented.

5.2.2 Problem decomposition

This approach is motivated by the observation that OD estimation, which is one of the two large components of this problem, can be written in an analytical form, which can be solved much more efficiently. A detailed description of the OD estimation process can be found in various sources, including: Cascetta et al. (1993); Ashok and Ben-Akiva (2000, 2002). A common approach is to use a linear analytical relationship, which can be solved using the Kalman Filter algorithm or GLS approaches.

In that case, it would be desirable to decompose the problem so that this analytical relationship can be exploited for demand parameter estimation. An example of such a decomposition strategy is presented in Figure 5-2. *A priori* demand and supply parameters are combined with available surveillance information to estimate the OD flows.

The updated demand —along with the *a priori* supply parameters and the surveillance information— can then be used as input for the estimation of the supply parameters . A

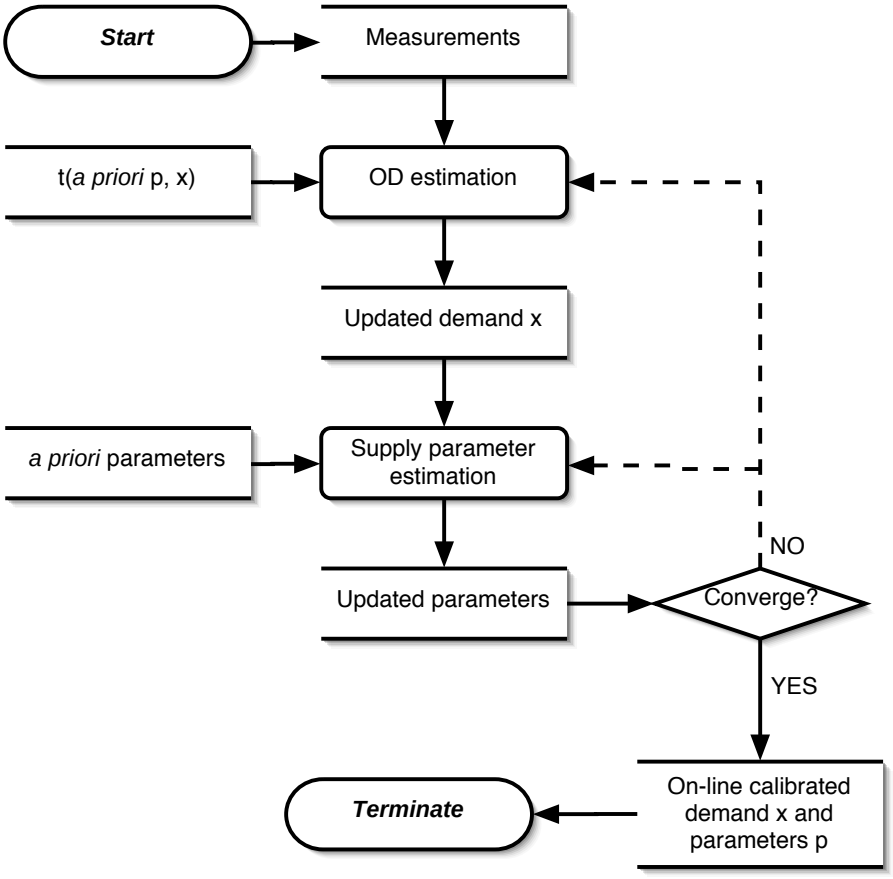

Figure 5-2: A decomposition strategy

modified version of the model shown in Section 3.4 will be used, the only difference being the state vector. In particular, the state vector would only include the supply parameters (speed–density relationship parameters and capacities). Therefore, since the dimension of the state is decreased, the number of (computationally expensive) function evaluations would be decreased proportionally.

Following the completion of these two steps, an updated set of demand and supply parameters would be available. The simulated network performance (using these inputs) can then be compared with the available surveillance information. If the simulated conditions match their observed counterparts adequately, then the algorithm terminates and these parameters can be used for network performance prediction. If, however, the convergence is not satisfactory, the algorithm can be repeated in an iterative fashion. In that case, the newly estimated demand and supply parameters can replace the *a priori* estimates as inputs and the OD estimation and supply parameter estimation steps can be repeated.

Each of these iterations, however, would involve the computationally intensive supply parameter estimation step. Another option could be to skip the supply parameter estimation step in the remaining iterations and keep the supply parameters fixed at the values obtained from the first (and only) supply parameter estimation iteration. For the additional iterations only the OD estimation would be performed.

The most computationally intensive component resulting from such a decomposition strategy would be —by far— the supply parameter estimation. In the absence of an analytical expression that would link the capacities to the measurements it would be necessary to use one of the algorithms presented in Chapter 4. The state vector would include the segment capacities and the speed–density relationship parameters.

Such decomposition approaches suffer from two main drawbacks. First, the decomposition of the problem into two separate demand and supply parameter estimation components would defeat the purpose of on–line calibration (i.e. jointly and simultaneously estimating demand and supply parameters). In fact, during the OD estimation all errors would be attributed to the OD flows (possibly overcompensating). When the supply parameter estimation is then performed, all errors would in turn be attributed to the supply parameters (while the algorithm would use the "wrong" estimated OD flows as input). Furthermore, the potential gains in terms of computation burden are limited by the ratio of the number of the supply

parameters to the total number of parameters to be estimated. As the demand and supply parameters are likely to be of the same order of magnitude, a speed–up factor of around 2 or 3 is expected (assuming no additional iterations involving the supply parameter estimation are performed).

An approach that can improve the overall runtime performance of either the EKF or the UKF algorithm at the expense of additional resources is presented in the following section.

5.2.3 Parallelization

Complex software systems involving multiple interacting components can often be sped up considerably by processing multiple tasks at a time *in parallel*. The degree to which such *parallelization* approaches can improve the computational properties of the overall system depend on the available hardware resources and the inherent delays in the system.

Delays in a parallel algorithm are generally due to two factors:

- Interprocess communication: As parts of a larger software system, parallel processes often need to share and exchange data. It is often the case that data generated by one task component needs to be transferred to another component for further processing, and then the re–processed data may be sent to a third component, and so on. Interprocess communication can add to the total runtime of a parallel algorithm if the amount of data that is transferred is large, and/or the data transfer between processes is slow.

- Idle processor time: A parallel algorithm would achieve optimal performance if all available processors were fully utilized all of the time. However, this is not always possible. Consider the situation described above, where one process has completed its task and is sending the output to another process that will use it for subsequent computations. Both processors are under–utilized during this transfer. A more serious delay may be sustained if a task depends on more than one other task. In that case, the processor cannot resume its operation until all "upstream" processes have finished.

Parallelization techniques are targeted at minimizing these two concerns.

Besides requiring a very similar number of function evaluations ($2n + 1$ for the EKF versus

$2n$ for the UKF), both processes share another common characteristic. Each function evaluation is totally independent from the others, i.e. the input to any of these function evaluations does *not* depend on the output of any of the other evaluations. A related consequence with significant practical implications is that the order in which the function evaluations are processed does not matter since subsequent computations cannot commence until all evaluations have been completed.

Another important observation is that all function evaluations take roughly the same time. This property can help minimize idle processor time and greatly simplifies parallelization. In particular, one of the main challenges in designing a parallel algorithm lies with the scheduling of the tasks and their assignment to the available processors. This can be a challenging task if the various processes have different runtimes (varying either among them, or from replication to replication). In the presented on–line calibration approach for DTA, however, each replication is one run of the simulator. Furthermore, the dimension and the characteristics of the problem are stable across replications. Therefore the runtime of each evaluation is known and roughly equal to all other evaluations.

Furthermore, there is a minimal level of interprocess communication required. In particular, it suffices to provide the simulator with a state vector to evaluate. When the evaluation is complete, only the generated measurement vector needs to be returned to the calling process for further processing. Therefore, a complete set of inputs for all function evaluations can be generated prior to their execution. Similarly, the remaining steps of each algorithm can be performed after the evaluations have been completed.

These three characteristics:

- Independence of runs
- Equal computation time per function evaluation, and
- Minimal interprocess communication

make both these algorithms very amenable to parallelization. Furthermore, the implementation of a parallel version of the algorithms is straightforward and simply involves allocating the function evaluation equally among the available processors. No complex parallelization algorithms or run–time optimization are required.

In the case of a simulation based DTA system, the cost of the function evaluations is so high that the remaining steps are negligible. Therefore, the performance of the algorithm is approximately inversely proportional to the number of available processors. Finally, it is worth noting that parallelizing the algorithms does not require any approximation.

5.2.4 Limiting Extended Kalman Filter

In this section, a special case of the Extended Kalman Filter is presented that significantly improves the computational performance of the algorithm. As mentioned in Section 4.3, the most *computationally intensive* step in the EKF algorithm is the linearization of the measurement equation (Equation 4.17), as it requires the use of numerical derivatives. Using central derivatives, $2\mathfrak{n}$ function evaluations are required, where $\mathfrak{n}$ is the size of the state vector. Another costly operation is the inversion required for the computation of the Kalman gain (Equation 4.18).

In real–time applications, it may be possible to replace the Kalman Gain matrix $\mathbf{G}_\mathfrak{h}$ by a constant gain matrix considerably decreasing the computation time. The *limiting* Kalman Filter will be defined by replacing $\mathbf{G}_\mathfrak{h}$ with its "limit" $\mathbf{G}$, called the *limiting* (or *stable*) *Kalman gain matrix* (Chui and Chen, 1999). The main steps of the Limiting Extended Kalman Filter (LimEKF) algorithm are presented in Algorithm 5.1. The differences from the EKF algorithm are limited to the computation of the numerical derivative (which is not computed on–line in the LimEKF) and the use of the *limiting* Kalman gain $\mathbf{G}$ for every iteration (Equations 5.18 and 5.19).

The limiting Kalman gain matrix can be computed off–line. The simplest way would be to express the limiting Kalman gain matrix as the average of a number of available Kalman gain matrices:

$$\mathbf{G} = \frac{\sum_{\mathfrak{m}=1:\mathsf{M}} \mathbf{G}_\mathfrak{m}}{\mathsf{M}} \tag{5.20}$$

where $\mathbf{G}_\mathfrak{m}$ is the Kalman gain obtained from EKF during interval $\mathfrak{m}$ and M is the total number of available Kalman gain matrices.

Several strategies can be developed to improve the quality of the limiting Kalman gain. For example, the EKF could be run off–line, with each run producing a new Kalman gain

Algorithm 5.1 Limiting Extended Kalman Filter

Generation of *limiting* Kalman gain matrix $\mathbf{G}$ and $\mathbf{H}$

Initialization

$$\mathbf{X}_{0|0} = \mathbf{X}_0 \tag{5.14}$$

$$\mathbf{P}_{0|0} = \mathbf{P}_0 \tag{5.15}$$

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

Time update

$$\mathbf{X}_{h|h-1} = \mathbf{F}_{h-1}\mathbf{X}_{h-1|h-1} \tag{5.16}$$

$$\mathbf{P}_{h|h-1} = \mathbf{F}_{h-1}\mathbf{P}_{h-1|h-1}\mathbf{F}_{h-1}^{T} + \mathbf{Q}_h \tag{5.17}$$

Measurement update

$$\mathbf{X}_{h|h} = \mathbf{X}_{h|h-1} + \mathbf{G}\left[\mathbf{Y}_h - \mathbf{h}\left(\mathbf{X}_{h|h-1}\right)\right] \tag{5.18}$$

$$\mathbf{P}_{h|h} = \mathbf{P}_{h|h-1} - \mathbf{GHP}_{h|h-1} \tag{5.19}$$

end for

matrix. These Kalman gain matrices could then be used to update the *limiting* Kalman gain matrix. Another strategy would be to consider only the last few Kalman gain matrices, i.e. use a type of moving average. Weighted averages (e.g. using lower weights for "older" gain matrices) can also be considered.

The main component of the Kalman gain matrix is the derivative $\mathbf{H}_h$ of the measurement equation. This is directly required in Equation 5.19. Using the same principle as above, it is possible to replace the time–dependent matrix $\mathbf{H}_h$ with the average $\mathbf{H}$ of a number of available matrices:

$$\mathbf{H} = \frac{\sum_{m=1:M} \mathbf{H}_m}{M} \tag{5.21}$$

where $\mathbf{H}_m$ is the matrix obtained from EKF during interval m and M is the total number of available matrix. The resulting matrix $\mathbf{H}$ can be then used to update the state covariance in Equation 5.19.

5.3 Conclusion

Practical considerations associated with the on–line calibration approach have been addressed. A staged approach that can be used if some of the model inputs are not available is presented. The missing information is generated from the output of a "warm–up" stage with limited data requirements.

Computational considerations are also discussed and alternative approaches to improve the runtime of the on–line calibration are presented. The advantages and drawbacks of each alternative are presented. Problem decomposition and the limiting variation of the EKF lead to approximations, while parallelization does not affect the accuracy of the model. In terms of computational performance, the limiting EKF algorithm is expected to provide the most dramatic improvements.

Chapter 6

Case study

Contents

In the previous chapters, an on–line calibration approach for Dynamic Traffic Assignment (DTA) has been developed, solution algorithms have been presented, and practical considerations have been addressed. The objective of this chapter is to demonstrate the approach.

In particular, the presented case study aims to achieve the following three **objectives**:

- **Demonstrate the feasibility of the approach**
  - The on–line calibration approach improves on the state–of–the–art by jointly estimating demand and supply parameters. The resulting formulation is flexible and general. However, since it also lacks an analytical relationship its solution requires numerical methods. It is necessary to demonstrate that the presented methodology can be successfully incorporated into a state–of–the–art DTA system. Furthermore, an application to a real network would highlight potential issues.
- **Verify the importance of on–line calibration**
  - The joint estimation of demand and supply parameters significantly increases the complexity of the on–line calibration problem. Therefore, in order to motivate its use, it is important to verify that it can provide significant benefits (over the "base–case" of only estimating OD flows on–line)
- **Test the candidate algorithms based on several criteria**
  - The ability of the algorithms to accurately estimate and predict traffic conditions is probably the most important benchmark
  - The computational properties of the candidate algorithms are also very important, since on–line calibration has strict computational constraints.
  - The algorithms should be robust, so that they can be transferred to different conditions.

The on–line calibration has been implemented and demonstrated as it applies to the DynaMIT–R DTA system. Three algorithms have been implemented (EKF, LimEKF and UKF) and their performance for a freeway network in Southampton, UK, is presented.

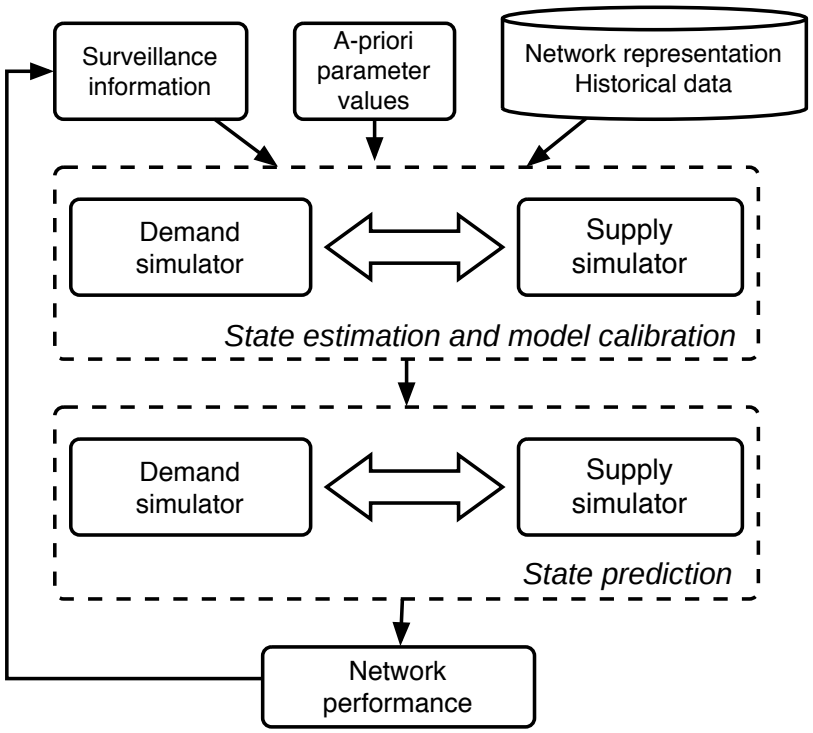

Figure 6-1: DynaMIT–R overview

The DynaMIT–R system is presented in the next section. A detailed presentation of the methodology used in this case study follows, including a description of the data, the experimental design, the measures of effectiveness, the off–line calibration, and several implementation details. Results of the case study are presented in detail. Finally, major findings of the case study are summarized.

6.1 The DynaMIT–R system

DynaMIT–R is a state–of–the–art DTA system (Figure 6-1). The high–level framework of DynaMIT–R has been presented in Section 1.2.

The key to the functionality of DynaMIT–R is its detailed network representation, coupled with models of traveler behavior. Through an effective integration of historical databases with real-time inputs, DynaMIT–R is designed to efficiently achieve:

- Real time estimation of network conditions.
- Rolling horizon predictions of network conditions in response to alternative traffic

control measures and information dissemination strategies.

- Generation of traffic information and route guidance to steer drivers towards optimal decisions.

To sustain users acceptance and achieve reliable predictions and credible guidance, DynaMIT–R incorporates unbiasedness and consistency into its core operations. Unbiasedness guarantees that the information provided to travelers is based on the best available knowledge of current and anticipated network conditions. Consistency ensures that DynaMIT–R's predictions of expected network conditions match what drivers would experience on the network. DynaMIT–R has the ability to trade-off level of detail (or resolution) and computational practicability, without compromising the integrity of its output.

Its important features include:

- A microscopic demand simulator that generates individual travelers and simulates their pre-trip and en-route decisions (choice of departure time and route) in response to information provided by available Advanced Traveler Information Systems (ATIS).
- Estimation and prediction of origin-destination flows.
- Simulation of different vehicle types and driver behaviors.
- A mesoscopic supply simulator that explicitly captures traffic dynamics related to the development and dissipation of queues, spillbacks, and congestion.
- Time–based supply simulation that simulates traffic operations at a user–defined level of detail that facilitates real–time performance. The level of detail could be determined by the choice of time steps and the level of aggregation of vehicles into homogeneous packets.
- Traveler information and guidance generation based on predicted traffic conditions to account for driver over-reaction to incident congestion. The system iterates between predicted network state, driver response to information and the resulting network state, towards the generation of consistent and unbiased information strategies.
- Adaptable to diverse ATIS requirements.

- Distinguishes between informed and uninformed drivers.
- Fuses historical, surveillance and O-D data to generate reliable OD estimates in real-time. The system records the results from previous O-D estimations to update O-D databases.
- Uses a rolling horizon to achieve efficient and accurate real-time estimations and predictions.
- Handles real–time scenarios including incidents, special events, weather conditions, highway construction activities and fluctuations in demand.
- Integrates with the MITSIMLab microscopic traffic simulator for offline evaluation and calibration.
- Ready for integration with external global systems (such as a TMC) using an external distributed CORBA interface, which allows for future adaptability and expansion.

DynaMIT–R is composed of several detailed models and algorithms to achieve two main functions:

- Estimation of current network state using both historical and real-time information.
- Generation of prediction-based information for a given time horizon.

The estimation and prediction phases operate over a rolling horizon.

The state estimation module provides estimates of the current state of the network in terms of O-D flows, link flows, queues, speeds and densities. This step represents an important function of DTA systems, since information obtained from the traffic sensors can vary depending on the type of surveillance system employed. In the presence of *floating* or *probe* vehicles in the network, or the existence of Automated Vehicle Identification Information (AVI) systems, detailed information about vehicle location and possibly origin and destination could be obtained. The DynaMIT–R system models can incorporate information obtained from such advanced surveillance systems (Antoniou et al., 2004).

While such systems are becoming available and may become ubiquitous in the future, most existing surveillance systems are based on vehicle detectors located at critical points in the

network. The information provided by these traffic sensors therefore must be used to infer traffic flows, densities and queue lengths in the entire network.

The main models used by the State Estimation module are:

- A demand simulator that combines real-time O-D estimation with user behavior models for route and departure time choice.
- A network state estimator (also known as the supply simulator) that simulates driver decisions and collects information about the resulting traffic conditions.

The demand and supply simulators interact with each other in order to provide demand and network state estimates that are congruent and utilize the most recent information available from the surveillance system (Figure 6-1).

Demand estimation in DynaMIT–R is sensitive to the guidance generated and information provided to the users, and is accomplished through an explicit simulation of pre-trip departure time, mode and route choice decisions that ultimately produce the O-D flows used by the O-D estimation model. The pre-trip demand simulator updates the historical O-D matrices by modeling the reaction of each individual driver to guidance information. The consequent changes are then aggregated to obtain updated historical O-D matrices.

However, these updated historical O-D flows require further adjustments to reflect the actual travel demand in the network. Reasons for the divergence of actual O-D flows from historical estimates include capacity changes on the network (such as the closure of roads or lanes), special events that temporarily attract a large number of trips to a destination, and other day-to-day fluctuations. Consequently, one of the requirements for dynamic traffic modeling is the capability to estimate (and predict) O-D flows in real time. The O-D model uses updated historical O-D flows, real–time measurements of actual link flows on the network, and estimates of assignment fractions (the mapping from O-D flows to link flows based on route choice fractions and travel times) to estimate the O-D flows for the current estimation interval.

The network state estimator utilizes a traffic simulation model that simulates the actual traffic conditions in the network during the current estimation interval. The inputs to this model include the travel demand (as estimated by the demand simulator), updated capacities and traffic dynamics parameters, the control strategies implemented and the

traffic information and guidance actually disseminated. The driver behavior model captures the responses to ATIS in the form of en–route choices.

One of the inputs to the O-D estimation model is a set of assignment matrices. These matrices map the O-D flows from current and past intervals to link flows in the current interval. The assignment fractions therefore depend on the time interval, and also on the route choice decisions made by individual drivers. The flows measured on the network are a result of the interaction between the demand and supply components. It may be necessary to iterate between the network state estimation and the OD estimation models until convergence is achieved. The output of this process is an estimate of the actual traffic conditions on the network, and information about origin-destination flows, link flows, queues, speeds and densities.

6.2 Case study methodology

6.2.1 Data description

The network

The network includes a 35km long part of freeway (M27) from Southampton, U.K. The network starts to the west of the city of Southampton, then goes around it, and continues eastbound towards Portsmouth (see Figure 6-2). The network also includes seven off–ramps and eight on–ramps.

A schematic representation is shown in Figure 6-3 which indicates the ten sensors which provide traffic information (counts, speeds and occupancies). Traffic is loaded onto the network via twenty origin-destination pairs. The peak period for this direction is the afternoon/evening.

Figure 6-2: The study area

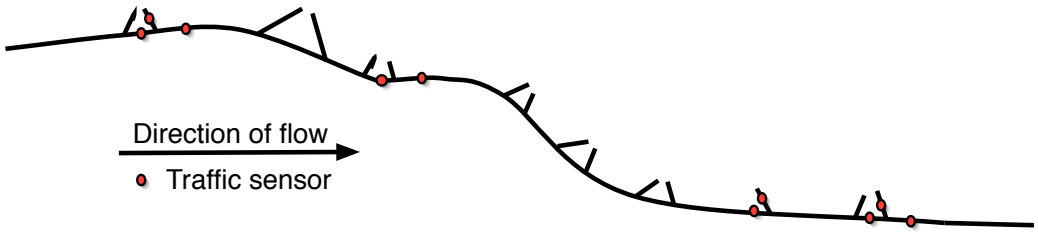

Figure 6-3: Schematic of the study network

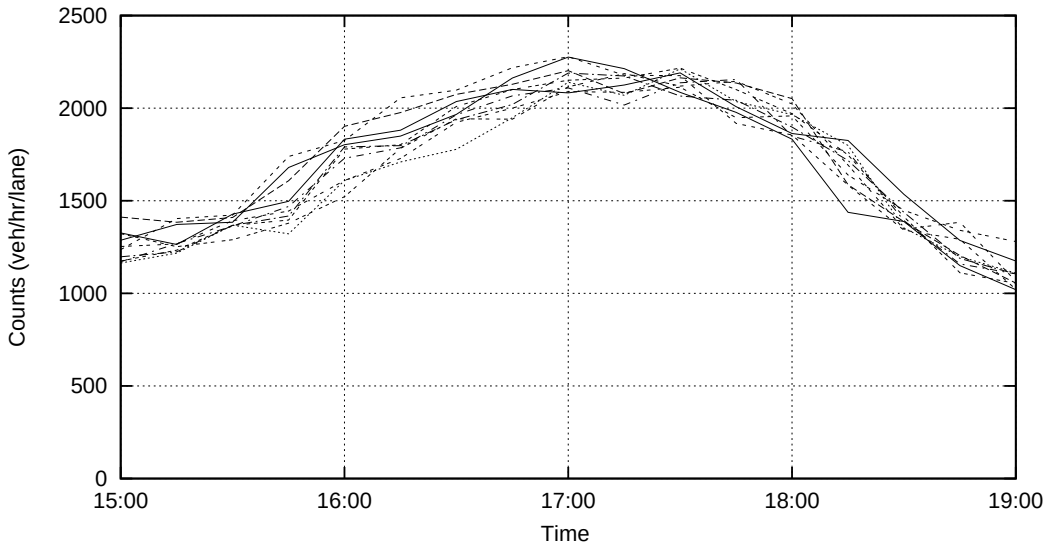

Figure 6-4: Mainline sensor counts distribution (all days)

Traffic characteristics

Weekday data from the first two weeks of September 2001 are used in this section to obtain some intuition about the traffic patterns in the network. Figure 6-4 shows the distribution of the counts observed at a mainline sensor. It appears that the traffic pattern is stable over days and traffic counts do not vary greatly by day.

Figures 6-5 and 6-6 show the speed and density distribution at the same mainline sensor. When the weather conditions are dry (6-5), the speed decreases from around 120kph to approximately 100kph. Under wet weather (6-6), however, speed drops to approximately 60kph.

6.2.2 Experimental design

The two dimensions (or factors) that are used in the experimental design are:

- Scope of the on–line calibration: i.e. whether only demand parameters are updated or whether both demand and supply parameters are updated jointly, and
- Type of day: for which the on–line calibration approach is applied. Following up on the discussion on the traffic characteristics (Section 6.2.1) we distinguish between two types of days based on the prevailing weather conditions: days with dry weather and

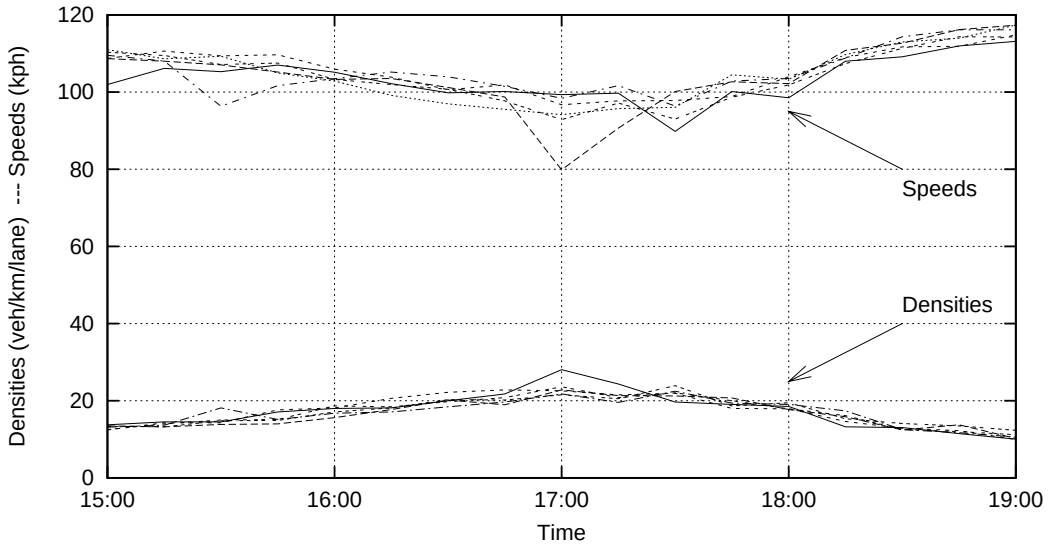

Figure 6-5: Speeds/densities (dry weather)

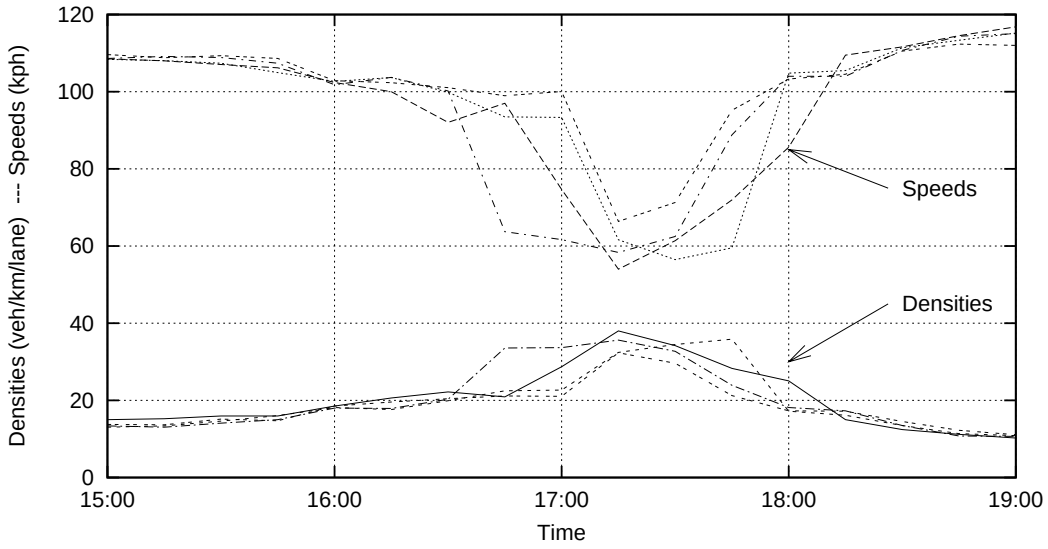

Figure 6-6: Speeds/densities (wet weather)

Table 6.1: Experimental design

Scope	Type of day: Day 1 (dry)	Type of day: Day 2 (wet)
Demand only (base)	KF/GLS	KF/GLS
Demand and supply	EKF LimEKF UKF	EKF LimEKF UKF

days with wet weather.

The dimensions and their levels are tabulated in Table 6.1. The situation when only the demand parameters are calibrated on–line is used as the base. Since this problem is the usual OD estimation problem, GLS or Kalman Filter algorithms can be applied.

When both the demand and supply parameters are jointly updated on–line, however, the problem cannot be represented analytically and the algorithms presented in earlier sections of this thesis can be used. For each type of day, the EKF, LimEKF, and UKF algorithms will be run.

The total number of experiments resulting from this design is eight.

6.2.3 Measures of effectiveness

One of the main outputs of DTA systems is traffic information and guidance, usually in the form of travel times. Speeds are the closest surveillance measurement and there are ways to compute travel times from speeds. Furthermore, given a properly calibrated traffic estimation and prediction system it is possible to obtain (simulated) travel times directly. On the other hand, the most ubiquitous traffic measurement is traffic counts.

Therefore, the two first measures of effectiveness are based on fit of estimated (predicted) speeds and counts with observed values, quantified using the normalized root mean square error (RMSN):

$$\mathsf{RMSN} = \frac{\sqrt{\mathsf{N}\sum_{\mathsf{N}}(\mathsf{y}-\hat{\mathsf{y}})^2}}{\sum_{\mathsf{N}}\mathsf{y}}$$

where N is the number of observations, y denotes an observation and $\hat{y}$ is the corresponding estimated (predicted) value.

The computational performance of the algorithms is another important consideration. In particular, given the *on–line* nature of the application, it is important to understand the computational complexity of each algorithm. Given a simulation–based DTA system, the function evaluations (required by the solution approaches) are by far the most computational intensive task, since each evaluation implies one run of the simulator. In the subsequent discussion, the number of function evaluations are used as a measure of effectiveness for each algorithm.

6.2.4 Off–line calibration

Since an off–line calibration was not available for this network, the first step in this case study was to perform an off–line calibration. Data from five weekdays with dry weather during the first two weeks of September 2001 were used for the off–line calibration.

A sequential calibration approach was followed. Supply parameters were first calibrated. Speed–density relationship parameters were obtained by fitting speed and density data to the appropriate functional form. Segment capacities were estimated using Highway Capacity Manual guidelines (HCM, 2000). Then, using the calibrated parameters as inputs, time–dependent OD flows were estimated. A detailed description of the calibration approach is available in Balakrishna (2002).

Speed–density relationship parameters were obtained by using non–linear regression to fit speed and density data to the speed–density relationship used by DynaMIT–R:

$$u = u_f \left[1 - \left(\frac{\max(0, K - K_{min})}{K_{jam}} \right)^{\beta} \right]^{\alpha} \tag{6.1}$$

where u denotes the speed, u_f is the free flow speed, K is the density, K_{min} is the minimum density, K_{jam} is the *jam* density and α and β are model parameters.

Using the above functional form, five parameters need to be estimated for each segment (using available observations for that segment alone). Furthermore, the addition of five parameters per segment would result in a fairly large number of parameters to estimate.

Therefore, it is often desirable (and a common practice) to classify segments with similar characteristics into homogeneous groups and estimate a single set of parameters for each segment type. The available data from all segments in a class could then be pooled and a single speed–density relationship estimated from the richer data. Only one set of parameters would then be added to the state vector for each segment type.

The 45 segments of the network were split into three groups with homogeneous characteristics. The mainline segments were classified into two types, while the ramp segments were grouped together.

Capacity computations are usually based on appropriate guidelines (e.g. the Highway Capacity Manual (HCM, 2000) for the United States). Although the study network is in the United Kingdom, no equivalent national guidelines are available for the United Kingdom and so, the Highway Capacity Manual guidelines were used. Analysis and comparison of the estimated capacities against observed counts were performed, to ensure that the capacity values did not result in counterintuitive results (such as observed sensor counts exceeding the segment capacity).

A sequential OD estimation approach (Balakrishna, 2002) was applied on five weekdays with dry weather. A static seed matrix was used to initialize the process. For the first day, the estimated OD flows from each interval were used as historical estimates for the next interval. The estimated flows for each day were then used as historical flows for the next day.

An ordinary least squares (OLS) approach was used for the first two days. At the end of the second day, measurement error covariances were estimates from the residuals of the fitted sensor counts and OD flows from their observed or historical values. This allowed for the use of a generalized least squares (GLS) approach for the remaining days. Estimated OD flows across time intervals were used to estimate autoregressive factors for the transition equation. The planning version of the DynaMIT system (DynaMIT–P) was used in this process.

Figures 6-7 through 6-10 graphically show the results of the off–line calibration process. The fit of the counts is shown in Figure 6-7 with observed counts (in vehicles per 15 minute interval) plotted on the x–axis, and simulated counts (in vehicles per 15 minute interval) plotted on the y–axis. A perfect fit would have all the points on the "45–degree line"

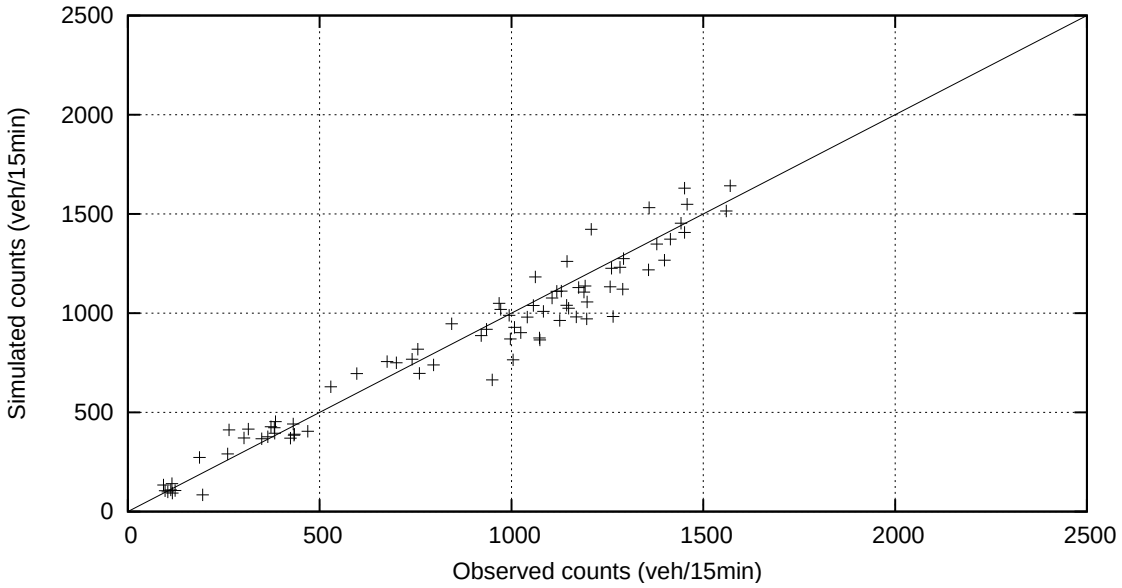

Figure 6-7: Off–line calibration results (counts)

(indicated by the solid diagonal line in the figure), meaning that the simulated counts perfectly matched the observed counts. The Normalized Root Mean Square error (RMSN) statistic for the counts was equal to 0.1232.

The fit of the speeds is shown in Figure 6-8. The total fit of the speeds, as quantified by the Normalized Root Mean Square error (RMSN) statistic, was equal to 0.1102. Furthermore, the observed and estimated speeds at a mainline and a ramp sensor are shown in Figures 6-9, and 6-10, respectively.

6.2.5 Implementation details

The on–line calibration approach was implemented in `octave` (a high–level language, primarily intended for numerical computations, `http://www.octave.org`) and `python` (an interpreted, interactive, object-oriented programming language, `http://python.org`). The supply simulator of DynaMIT–R was used for the function evaluations.

`octave` was used for the implementation of the three algorithms (EKF, UKF, and LimEKF). `python` was used primarily for the coordination of the function evaluations and the exchange of information between `octave` and DynaMIT–R (in particular preparation of inputs and processing of outputs).

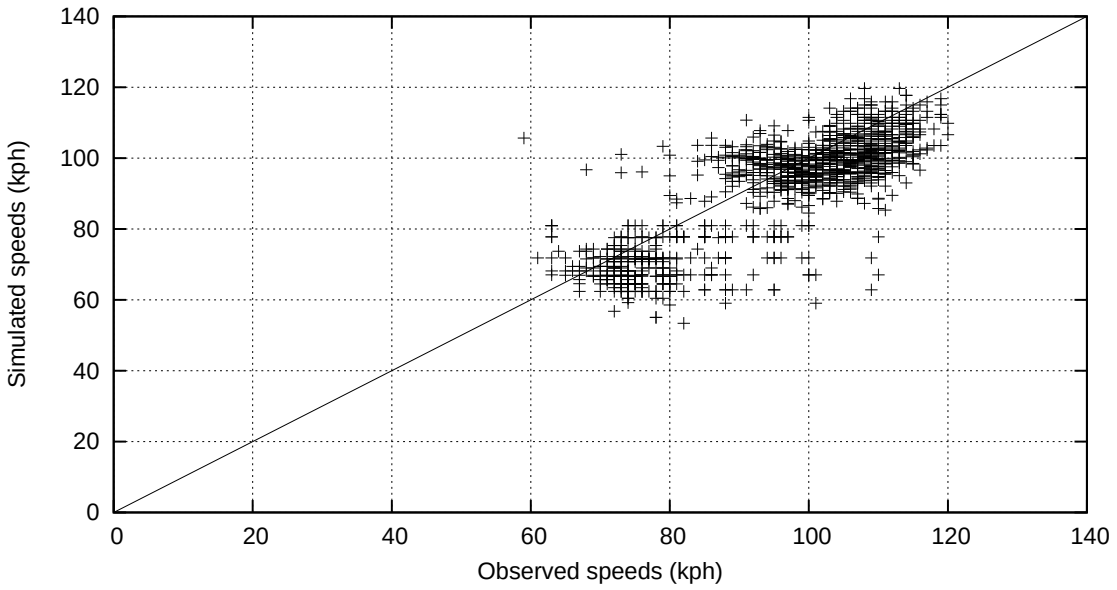

Figure 6-8: Off–line calibration results (speeds)

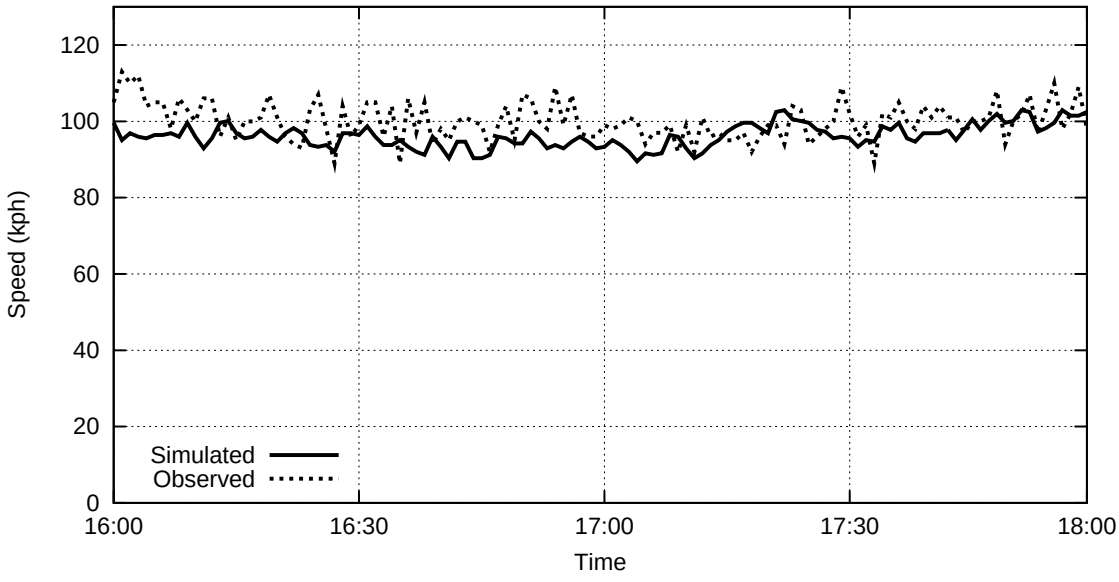

Figure 6-9: Off–line calibrated speeds – Mainline sensor

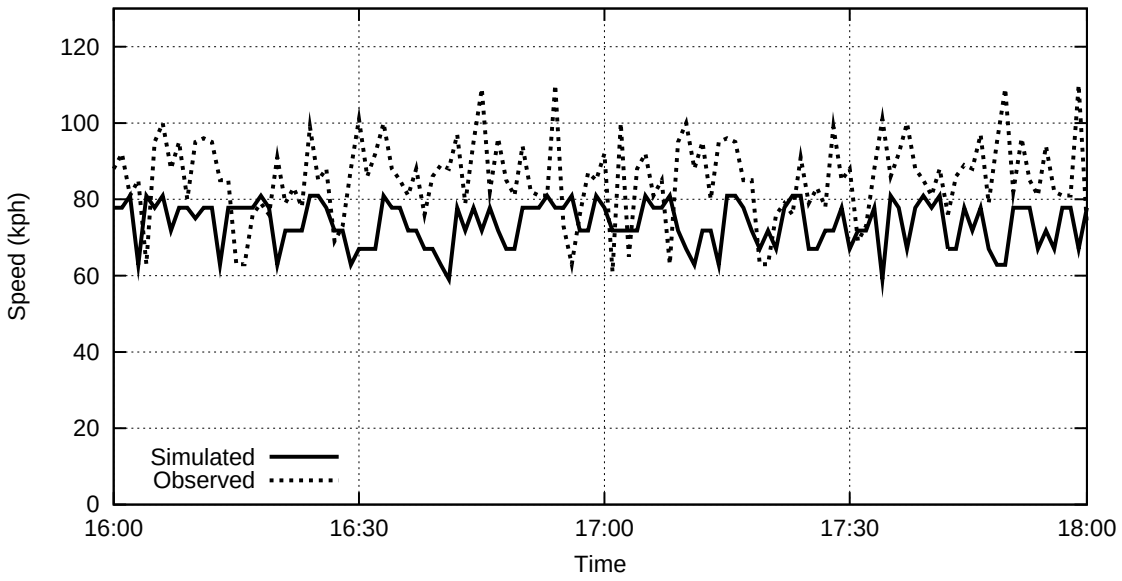

Figure 6-10: Off–line calibrated speeds – Ramp sensor

The state vector consists of OD flows, segment capacities and speed–density relationship parameters. The total dimension is 80, broken down in:

- 20 OD flows
- 45 segment capacities
- 15 speed–density relationship parameters: as mentioned before, segments have been grouped into three types, based on their geometric characteristics. A speed–density relationship (and therefore 5 parameters: free flow speed, minimum and jam density and exponents α and β) has been defined for each of the three segment types.

As mentioned earlier, two days (one with dry and one with wet weather conditions, different that those used for the off–line calibration) were used. The duration of the estimation and prediction intervals was set to fifteen minutes. OD estimation requires that the count measurements are aggregated to the duration of the interval (i.e. fifteen minutes in this case study). To maintain consistency between the various algorithms, this level of aggregation has been maintained for the counts in the on–line calibration approach. Furthermore, minute–by–minute speed and density surveillance information was used.

Therefore, the measurement vector for each fifteen–minute interval consisted of 390 elements:

- 15–minute count measurements: 10 count measurements in total
- Minute–by–minute speed measurements: 150 speed measurements
- Minute–by–minute density measurements: 150 density measurements
- *A priori* state vector: 80 elements comprising 20 OD flows, 45 capacities, and 15 speed–density relationship parameters (3 groups of 5 parameters each).

The transition fractions that were estimated during the off–line calibration were used for the OD flows. The degree of the autoregressive process for the OD flows was found to be equal to two. For the supply parameters, a degree of one was used for the autoregressive process. Furthermore, variance/covariance matrices were estimated based on the output of the off–line calibration.

The period of analysis comprises six intervals of fifteen minutes (i.e. from 16:15 to 17:45). A warm–up period of 75 minutes (from 15:00 to 16:15) is used to ensure that the network is adequately loaded.

6.3 Results

This section presents the case study results. Summary results are presented first to provide an overview of the value of on–line calibration. The performance of the candidate algorithms is also succinctly presented.

Interval–by–interval results are presented next. These results provide useful intuition into the performance of the on–line calibration approach at a finer scale.

Finally, the impact of the on–line calibration on the model parameters is demonstrated.

6.3.1 Summary results

Summary results for the dry day are presented in Table 6.2. This table summarizes the RMSN for estimated and predicted speeds and counts. The *base* row provides the performance when only demand parameters are estimated on–line. The next three rows show the results obtained when demand and supply parameters are jointly estimated (each row corresponds to one of the considered algorithms).

Table 6.2: Summary results (RMSN, dry weather)

	Estimated		One-step pred.		Two-step pred.		Three-step pred.	
Algorithm	speeds	counts	speeds	counts	speeds	counts	speeds	counts
Base	0.1266	0.1286	0.1494	0.1540	0.1448	0.1666	0.1494	0.1905
EKF	0.1107	0.1039	0.1209	0.1318	0.1303	0.1550	0.1331	0.2008
LimEKF	0.1121	0.1091	0.1249	0.1321	0.1346	0.1702	0.1362	0.2036
UKF	0.1156	0.1293	0.1249	0.1505	0.1315	0.1756	0.1346	0.2221

Table 6.3: Summary results (% improvement, dry weather)

	Estimated		One-step pred.		Two-step pred.		Three-step pred.	
Algorithm	speeds	counts	speeds	counts	speeds	counts	speeds	counts
Base RMSN	0.1266	0.1286	0.1494	0.1540	0.1448	0.1666	0.1494	0.1905
EKF	12.6%	19.2%	19.1%	14.4%	10.0%	7.0%	10.9%	−5.4%
LimEKF	11.5%	15.1%	16.4%	14.2%	7.0%	−2.2%	8.8%	−6.9%
UKF	8.7%	−0.6%	16.4%	2.3%	9.2%	−5.4%	9.9%	−16.6%

Table 6.3 presents the same results in a different way. In particular, the results for the cases where demand and supply parameters are estimated jointly are shown as percent improvement over the *base* case (i.e. when only demand parameters are estimated on–line). These results are also presented graphically in Figure 6-12 (estimated and predicted counts) and Figure 6-11 (estimated and predicted speeds).

These results indicate that the joint calibration of demand and supply parameters can improve the ability of the system to accurately estimate and predict the traffic conditions. The EKF algorithm exhibits the best performance with considerable improvements in estimation and prediction accuracy (except for three–step predicted counts).

A small decrease in performance (compared to the EKF) —but still a clear improvement— is obtained when the LimEKF algorithm is used. It is interesting to note that while the LimEKF algorithm has order(s) of magnitude lower computational complexity, it still provides a significant improvement over the base case. Improvements of more than 10% are obtained in estimated and one–step predicted speeds and counts. Furthermore, the LimEKF algorithm provides a 7% improvement in two–step predicted speeds and close to a 0% improvement in three–step predicted speeds. However, there is a small deterioration (−2.2%) in the two–step predicted counts and almost a 7% deterioration in three–step predicted counts.

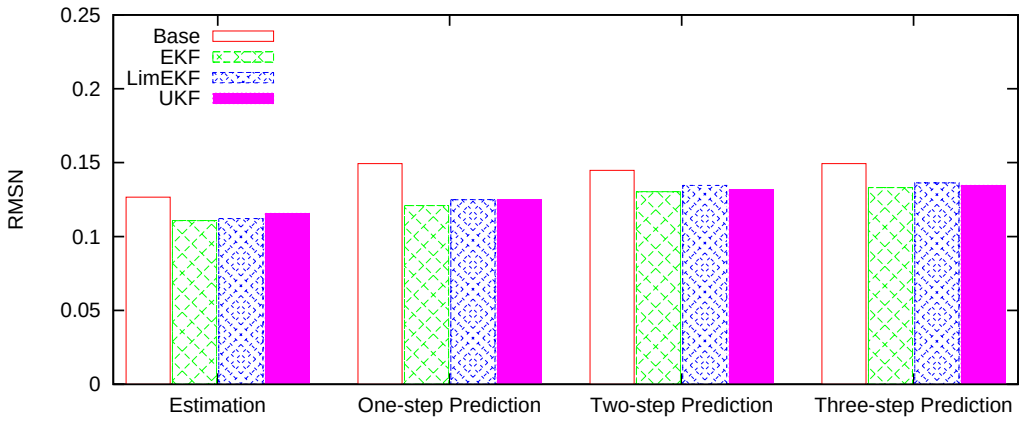

Figure 6-11: Summary statistics for estimated and predicted speeds (dry weather)

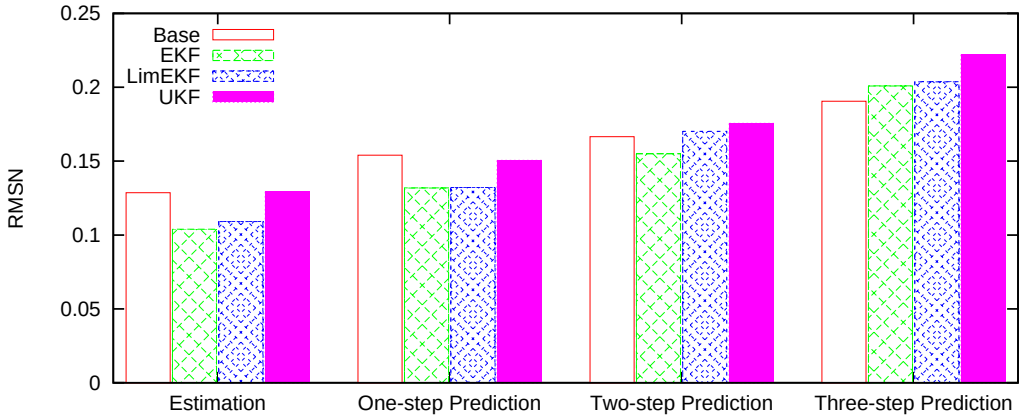

Figure 6-12: Summary statistics for estimated and predicted counts (dry weather)

The UKF algorithm seems to have the least desirable performance. While in general this algorithm provides improvement over the base case, its two–step and three–step predicted counts also deteriorate (-5.4% and -16.6%). Furthermore, (with the exception of two–step and three–step predicted speeds) this algorithm is generally outperformed by the LimEKF, which has vastly better computation properties.

Similar summary results for the wet day are presented in Tables 6.4 and 6.5 and Figures 6-13 and 6-14.

The results are similar to those obtained under dry weather. The EKF algorithm provides the best overall performance, with steady improvements for estimated and predicted speeds and counts. As a matter of fact, in this case improvement of more than 15% was obtained (with the exception of three–step prediction where an improvement of more that 10% was

Table 6.4: Summary results (RMSN, wet weather)

	Estimated		One-step pred.		Two-step pred.		Three-step pred.	
Algorithm	speeds	counts	speeds	counts	speeds	counts	speeds	counts
Base RMSN	0.1312	0.1154	0.1550	0.1334	0.1669	0.1388	0.1751	0.1424
EKF	0.1094	0.0946	0.1283	0.1096	0.1413	0.1149	0.1548	0.1425
LimEKF	0.1175	0.1023	0.1340	0.1250	0.1447	0.1370	0.1562	0.1663
UKF	0.1137	0.1129	0.1202	0.1268	0.1386	0.1536	0.1515	0.1787

Table 6.5: Summary results (% improvement, wet weather)

	Estimated		One-step pred.		Two-step pred.		Three-step pred.	
Algorithm	speeds	counts	speeds	counts	speeds	counts	speeds	counts
Base RMSN	0.1312	0.1154	0.1550	0.1334	0.1669	0.1388	0.1751	0.1424
EKF	16.6%	18.0%	17.2%	17.9%	15.3%	17.2%	11.6%	−0.1%
LimEKF	10.4%	11.4%	13.5%	6.3%	13.3%	1.3%	10.8%	−16.8%
UKF	13.3%	2.2%	22.5%	4.9%	17.0%	−10.7%	13.5%	−25.5%

obtained for the speeds, while the predicted counts' performance was practically the same)..

The LimEKF algorithm provides somewhat lower benefits than the EKF (with an improvement of more than 10% for predicted speeds). However, given the excellent computational properties of this algorithm, these benefits are very promising. The three–step predicted counts, however, show a significant deterioration (more than 16%.

While the UKF algorithm provides significant benefits in terms of speeds (even better than the other two algorithms for predicted speeds), the performance in terms of counts is not as good, with small benefits for estimated and one–step predicted counts, but significant disbenefits for two–step predicted (−10.7%) and three–step predicted (−25.5%) counts.

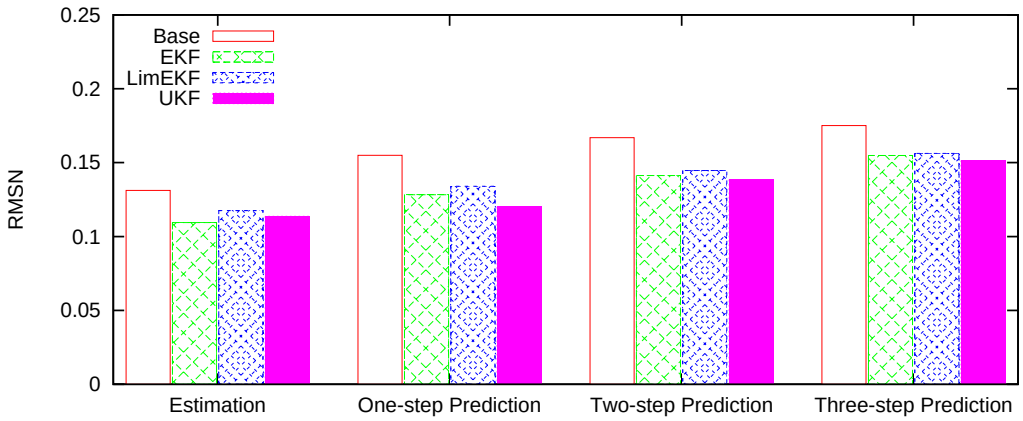

Figure 6-13: Summary statistics for estimated and predicted speeds (wet weather)

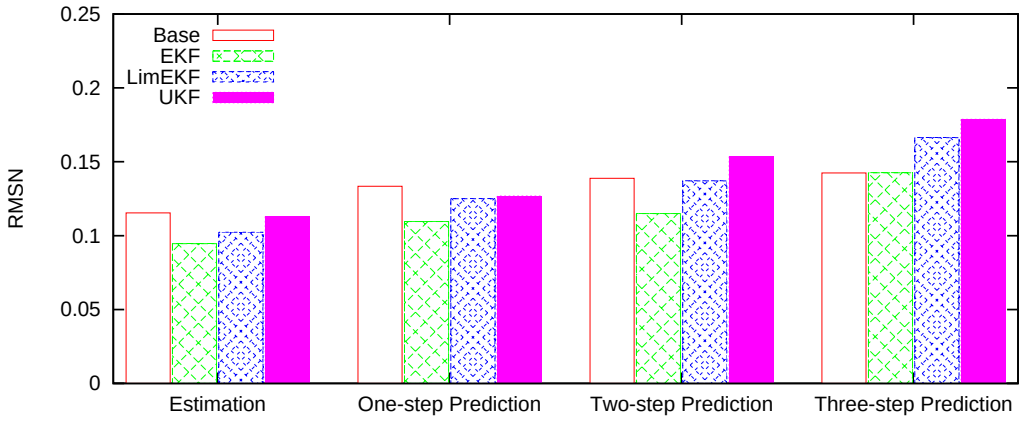

Figure 6-14: Summary statistics for estimated and predicted counts (wet weather)

The third measure of effectiveness (besides the fit of speed and counts) is the computational complexity of each algorithm. As discussed in Section 6.2.3, the computational performance of the on–line calibration approach for a simulation–based DTA system is determined largely by the number of function evaluations required. The EKF and the UKF algorithms require $2n$ and $2n+1$ function evaluations (and thus runs of the simulator) respectively (where n is the dimension of the state vector). Therefore, the two algorithms have similar computational requirements (converging as the dimension of the problem increases).

The LimEKF algorithm, on the other hand, requires a single function evaluation irrespective of the dimension of the problem. Therefore, the computational performance of this algorithm is vastly superior to that of the other two algorithms (EKF and UKF). The *constant* computational requirements of the LimEKF algorithm (i.e. the fact that a single function

evaluation is required irrespective of the application) makes it easy to obtain some further insight into the scalability of the approach to larger networks.

In particular, assuming that an estimation interval of fifteen minutes is used, this approach can be used for any application (i.e. combination of DTA system and network) that allows for one function evaluation (for the on–line calibration) and another run of the simulator (for the prediction of the state using the on–line calibrated parameters within that time–frame). A related observation is that using the LimEKF and the on–line calibration approach to jointly estimate all inputs and model parameters is computationally equivalent to the base case (i.e. only using OD estimation to calibrate OD flows on–line).

6.3.2 Results by time–interval

In the previous section, the performance of the on–line calibration approach for the entire study period was presented. The purpose of this section is to analyze the results interval–by–interval in order to obtain further insight.

Figures 6-15 through 6-18 show the estimated and predicted speeds by interval for dry weather. All algorithms show similar estimation performance with benefits between 16:15 and 17:30 and practically identical performance in the last time interval (17:30-17:45) (Figure 6-15). All algorithms show improvements for one–step predicted speeds (Figure 6-16). Benefits for most of the two–step and three–step predicted speeds are also obtained, except for the 17:30-17:45 interval (Figure 6-17).

Figures 6-19 through 6-22 show the estimated and predicted counts by interval for dry weather. Except for significant benefits during the interval 17:15–17:30, the performance of the EKF seems to be in general on par with the base case (in terms of estimated counts). This can be attributed to the fact that (for these intervals) the base algorithm already provides a reasonably good fit. However, during the time–intervals that the base algorithm provides poorer performance (17:00–17:15, and —in particular— 17:15–17:30), the EKF manages to provide a fit for the counts comparable to that obtained during the other intervals. Similar trends are exhibited in the counts predicted by the EKF, with the results deteriorating for three–step prediction.

This can be attributed to the extended objective function of the on–line calibration ap-

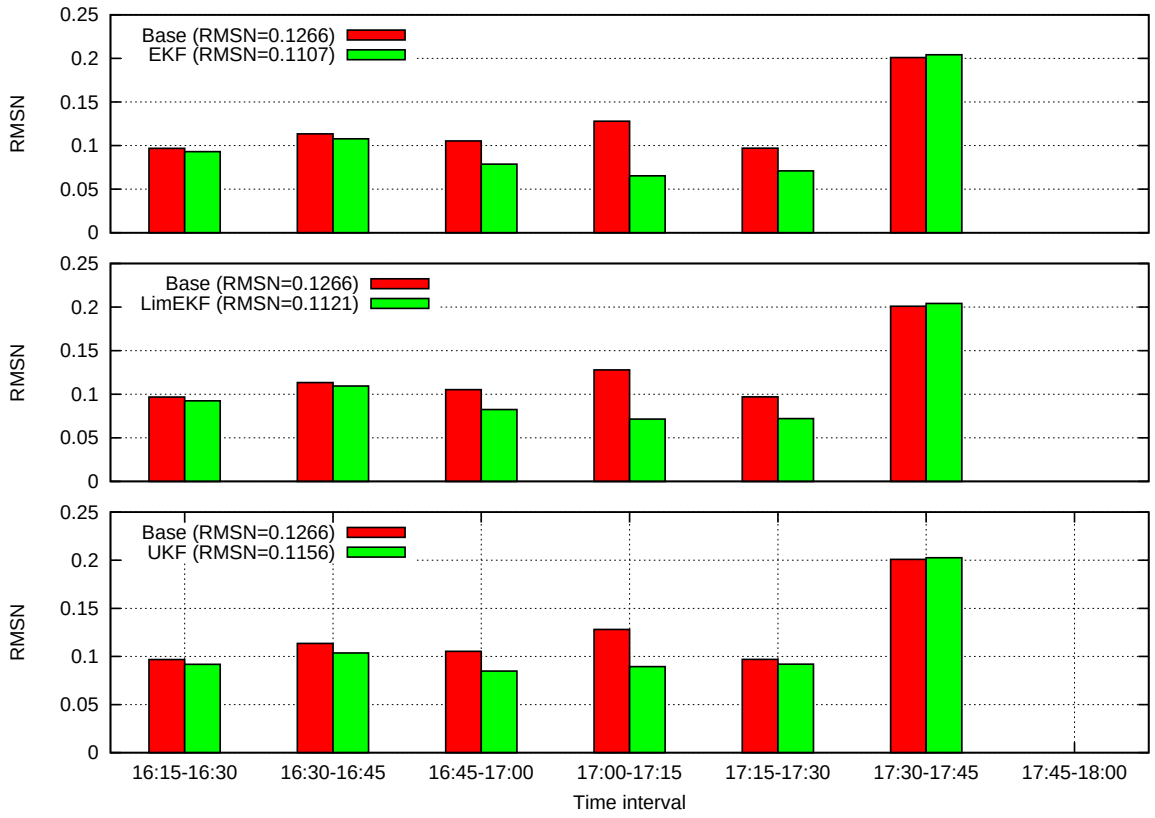

Figure 6-15: Statistics for estimated speeds by interval (dry weather)

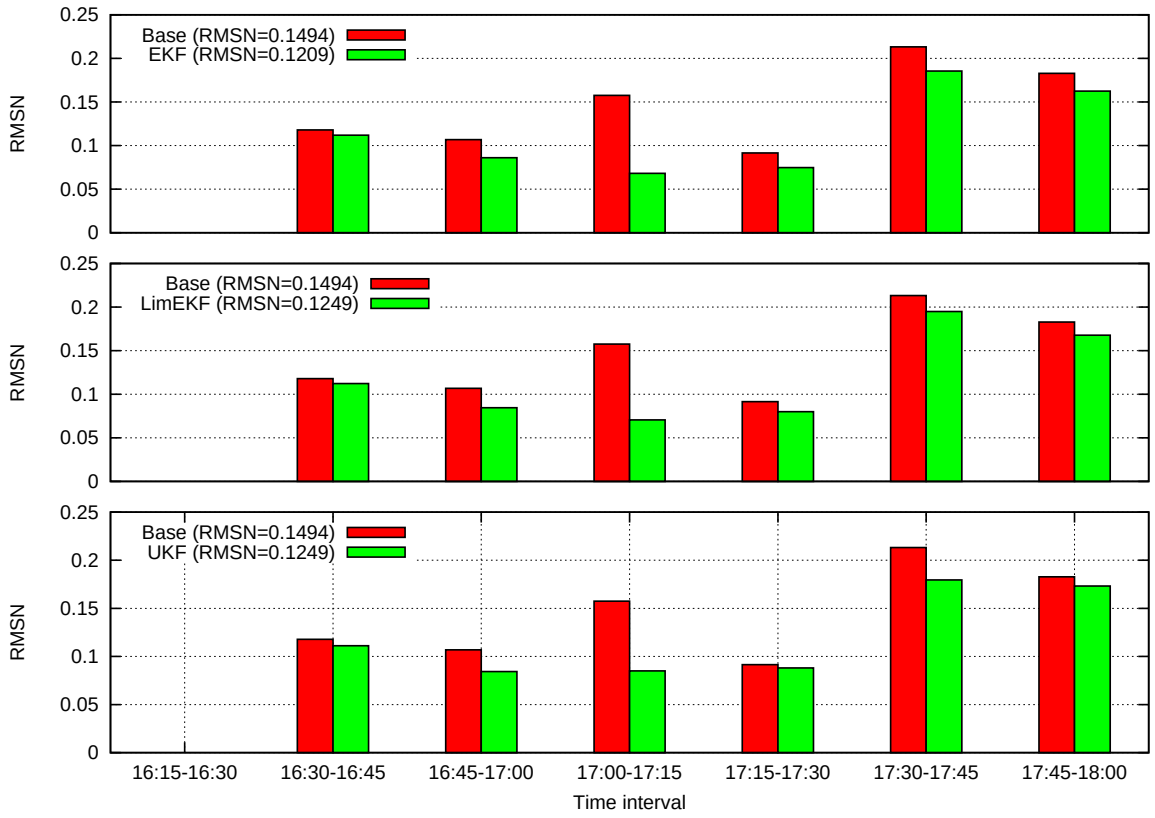

Figure 6-16: Statistics for one–step predicted speeds by interval (dry weather)

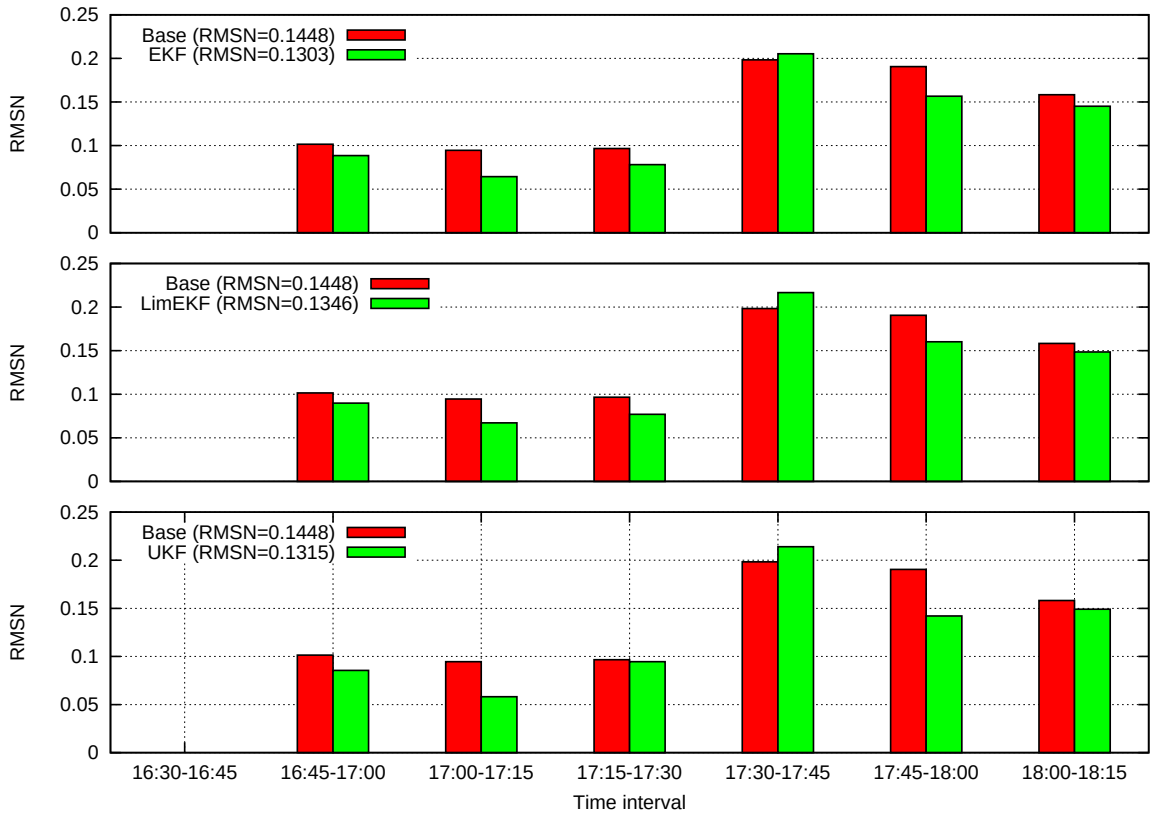

Figure 6-17: Statistics for two–step predicted speeds by interval (dry weather)

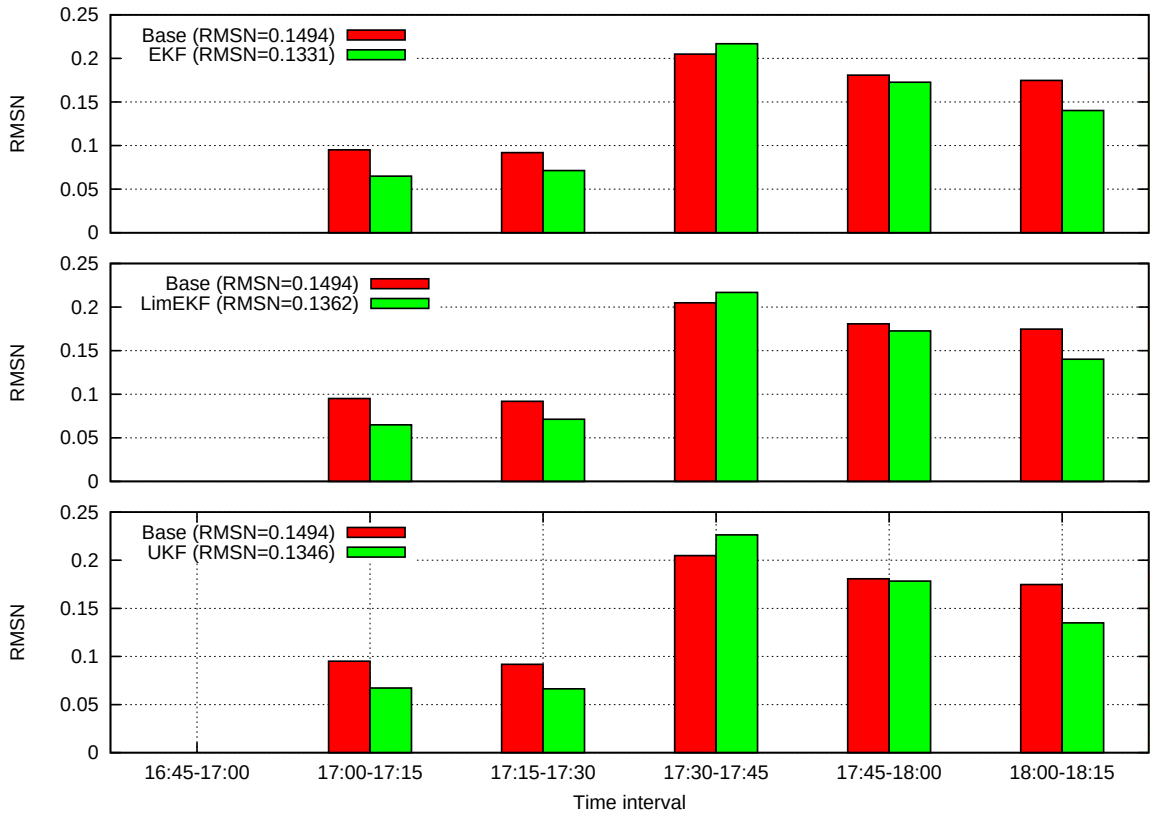

Figure 6-18: Statistics for three–step predicted speeds by interval (dry weather)

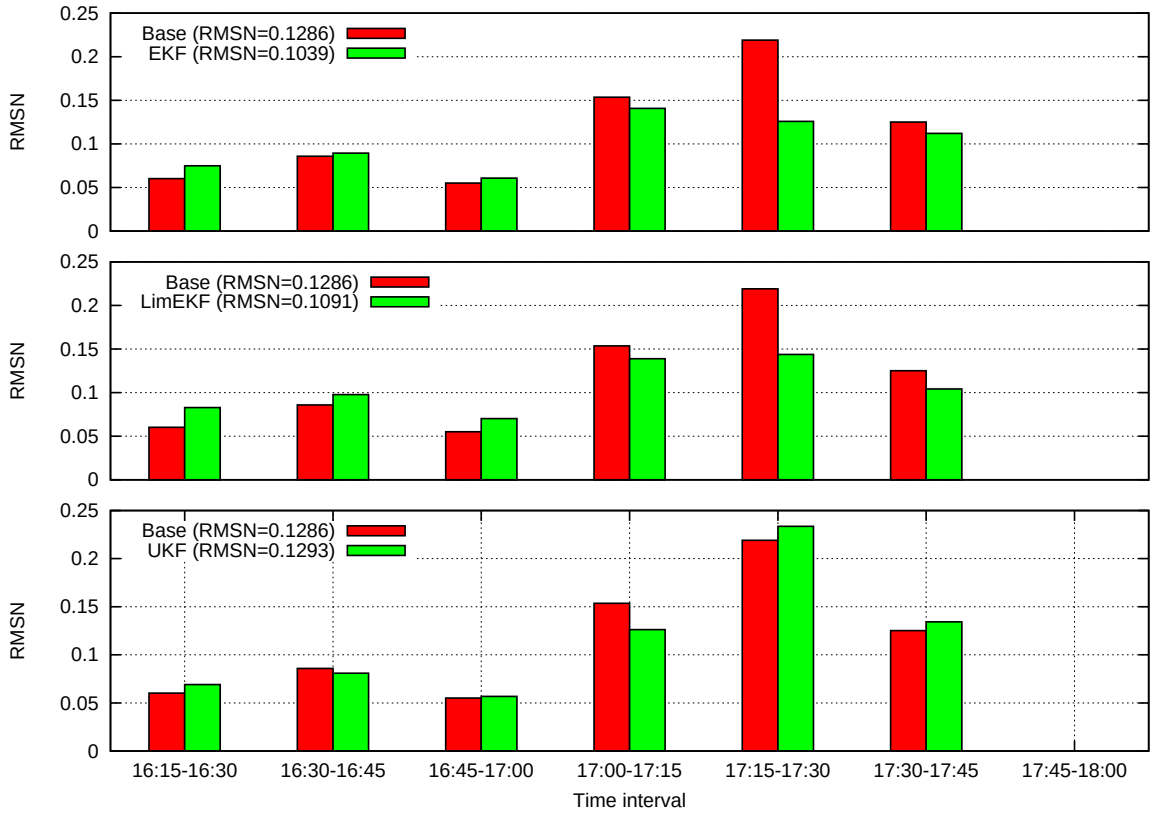

Figure 6-19: Statistics for estimated counts by interval (dry weather)

proach. Unlike the OD estimation problem (where link counts are used as measurements), the on–line calibration allows for easy incorporation of any other available type of information. For example, in this application, speeds and densities have been incorporated. Therefore, the objective function now includes both the deviations of estimated/predicted speeds and counts (from their observed values). Therefore, the algorithm is willing to trade–off some of the fit in the counts to gain accuracy in estimated/predicted speeds. This is a desirable property and one that results in the model capturing prevailing traffic conditions more accurately.

The results obtained with the LimEKF algorithm are similar to those obtained from the EKF. This is expected, since the LimEKF uses a Kalman gain that is based on a number

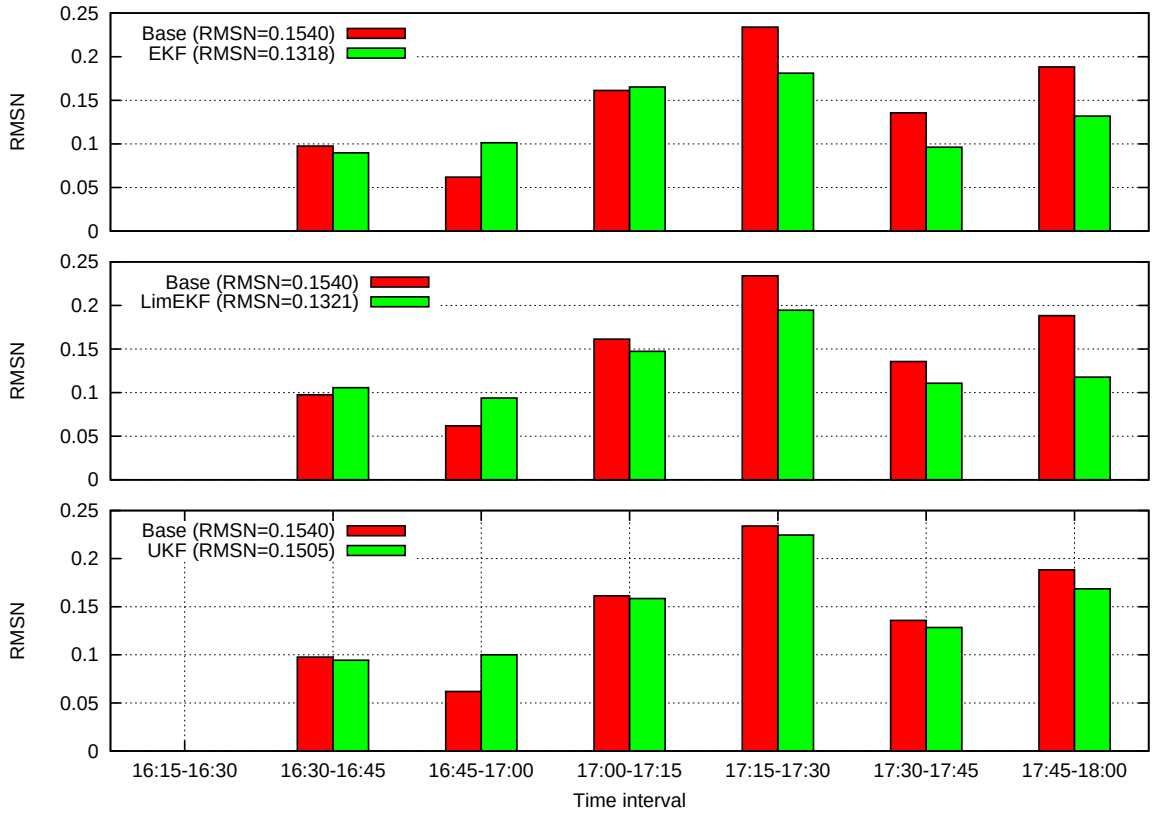

Figure 6-20: Statistics for one–step predicted counts by interval (dry weather)

of Kalman gains computed off–line using the EKF. Therefore, the structure of the matrix has similar characteristics. An additional benefit of using such a composite Kalman gain is that it encompasses more information (i.e. the information contained in the individual Kalman gain matrices). It is worth noting once more, that this is a very important result, since the LimEKF algorithm has very favorable computational properties.

It should be noted that even this algorithm does provide significant benefits in terms of speed estimation and prediction, while maintaining (on average) a similar performance in terms of counts. As suggested by the summary results, the UKF does not provide significant benefits over the base case in terms of counts estimation. Furthermore, the performance of the UKF does not show a trend similar to that exhibited by the EKF and LimEKF

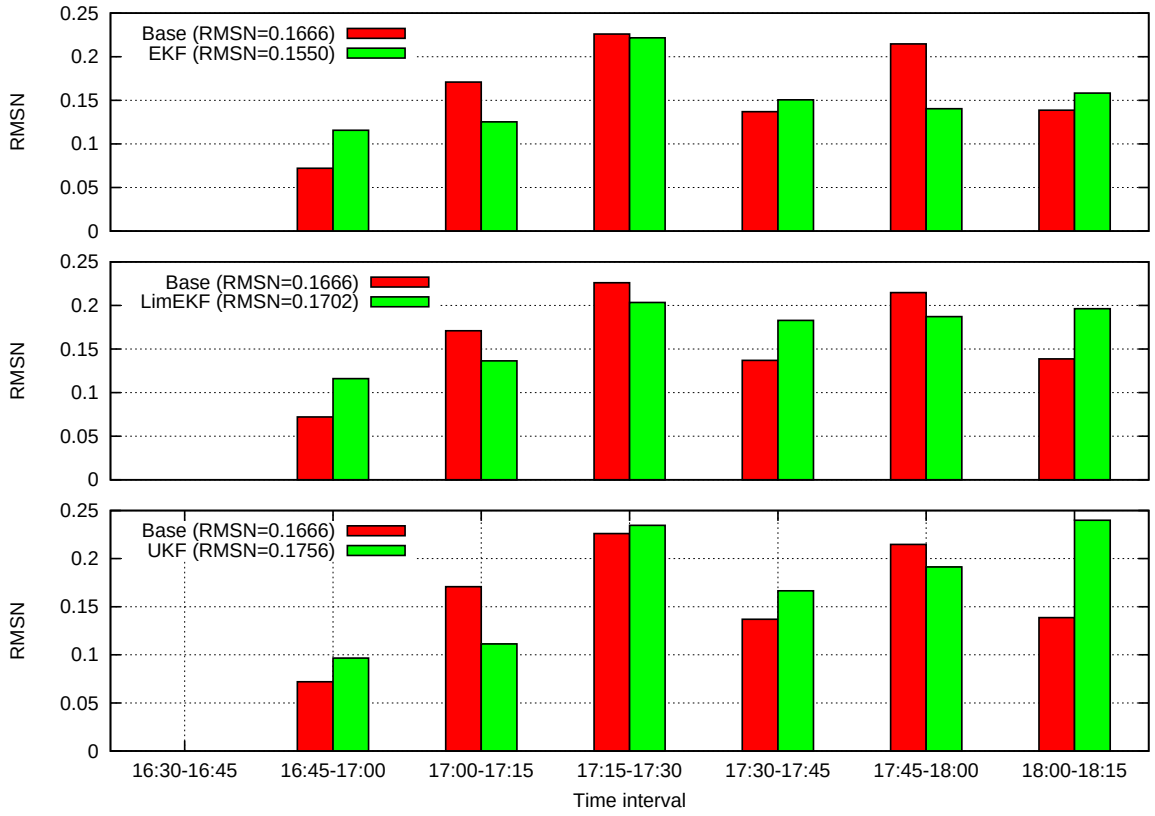

Figure 6-21: Statistics for two–step predicted counts by interval (dry weather)

algorithms (i.e. the reduction of "high" RMSNs).

Estimated and predicted speeds by interval for the wet day are shown in Figures 6-23 through 6-26 with similar results to those under dry weather. Figures 6-27 through 6-30 show the estimated and predicted counts by interval for wet weather again with similar performance to that observed under dry weather. In particular, the EKF algorithm provides benefits except for the predicted counts for the 16:45–17:00 interval. This is a phenomenon similar to that observed under dry weather for the same interval. Again, this can be attributed to the fact that the EKF may trade–off some of the fit in the counts in order to improve the fit in the speeds, thus capturing the overall traffic conditions more accurately.

The LimEKF provides performance comparable to that of the EKF, at only a small fraction

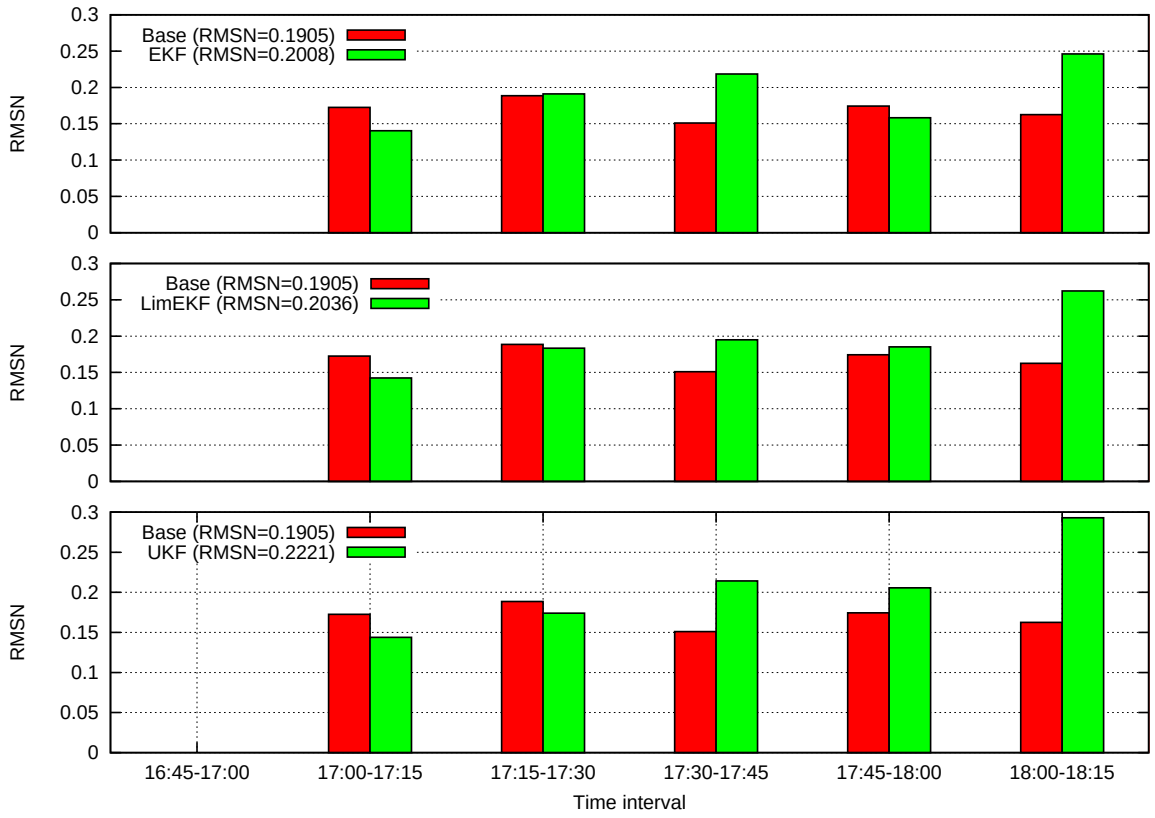

Figure 6-22: Statistics for three–step predicted counts by interval (dry weather)

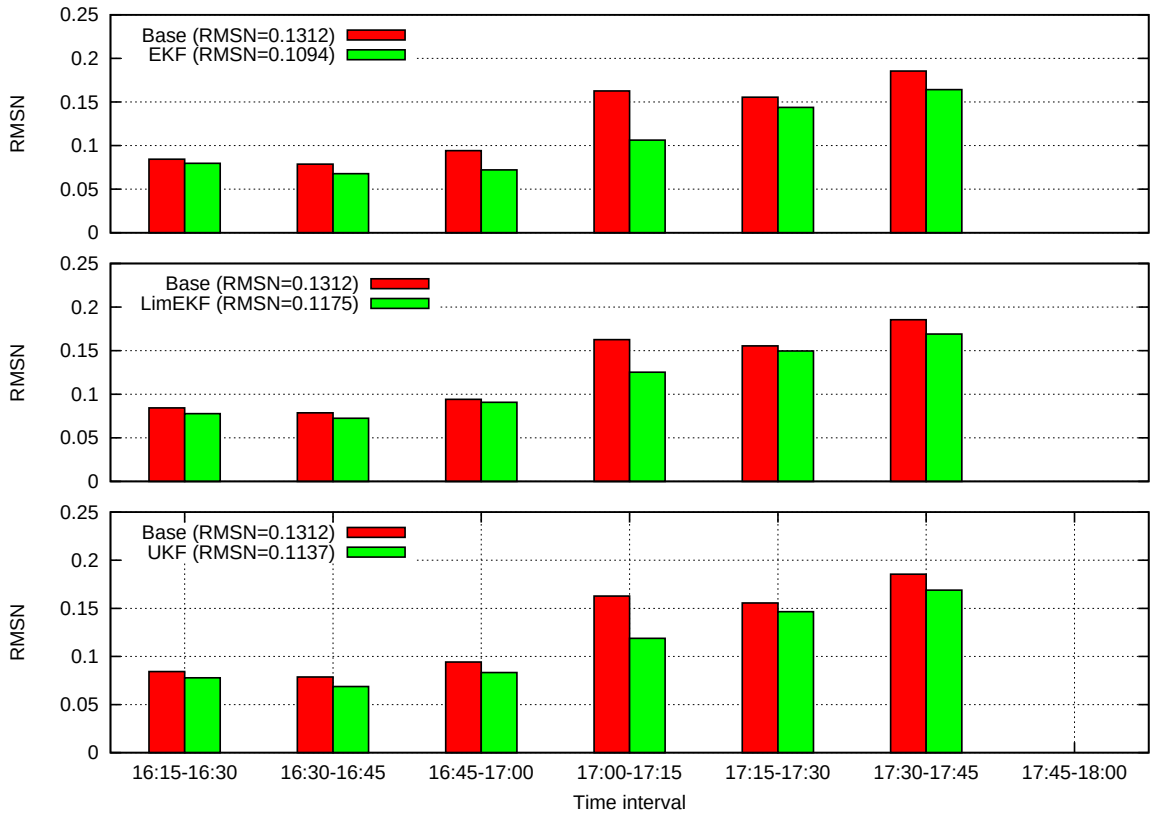

Figure 6-23: Statistics for estimated speeds by interval (wet weather)

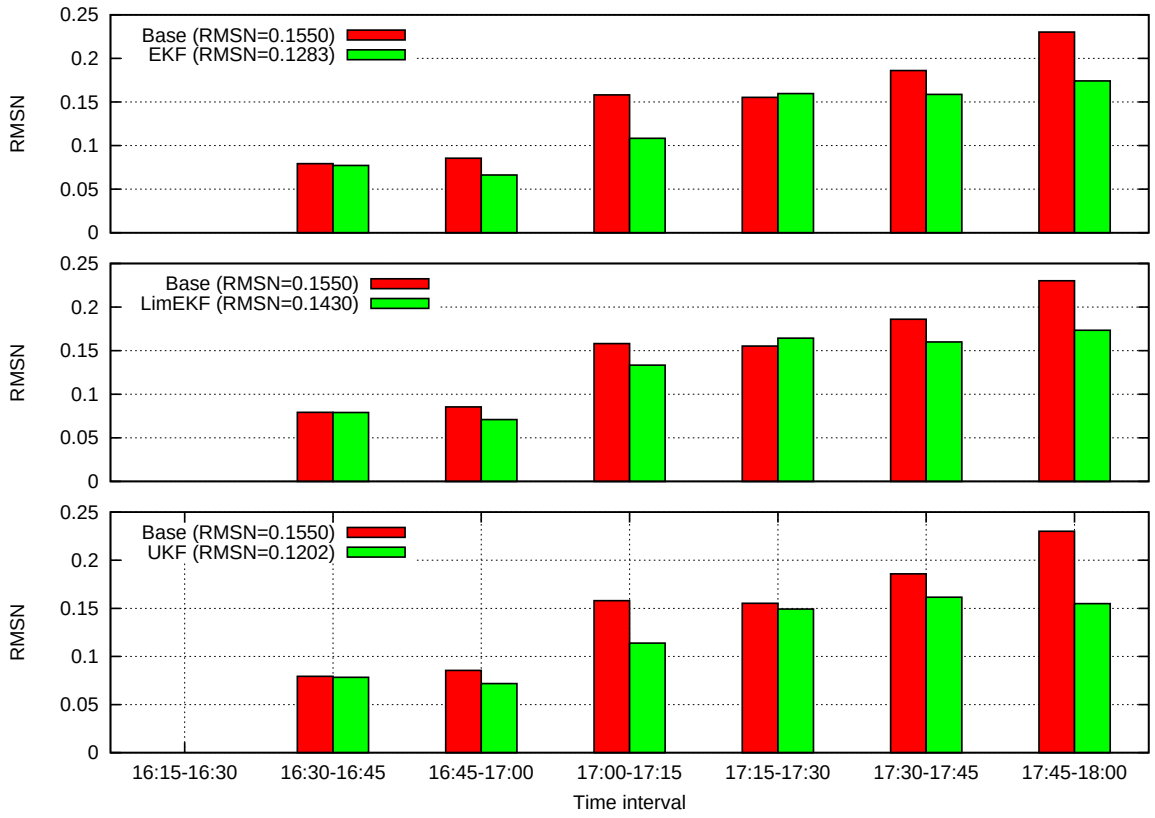

Figure 6-24: Statistics for one–step predicted speeds by interval (wet weather)

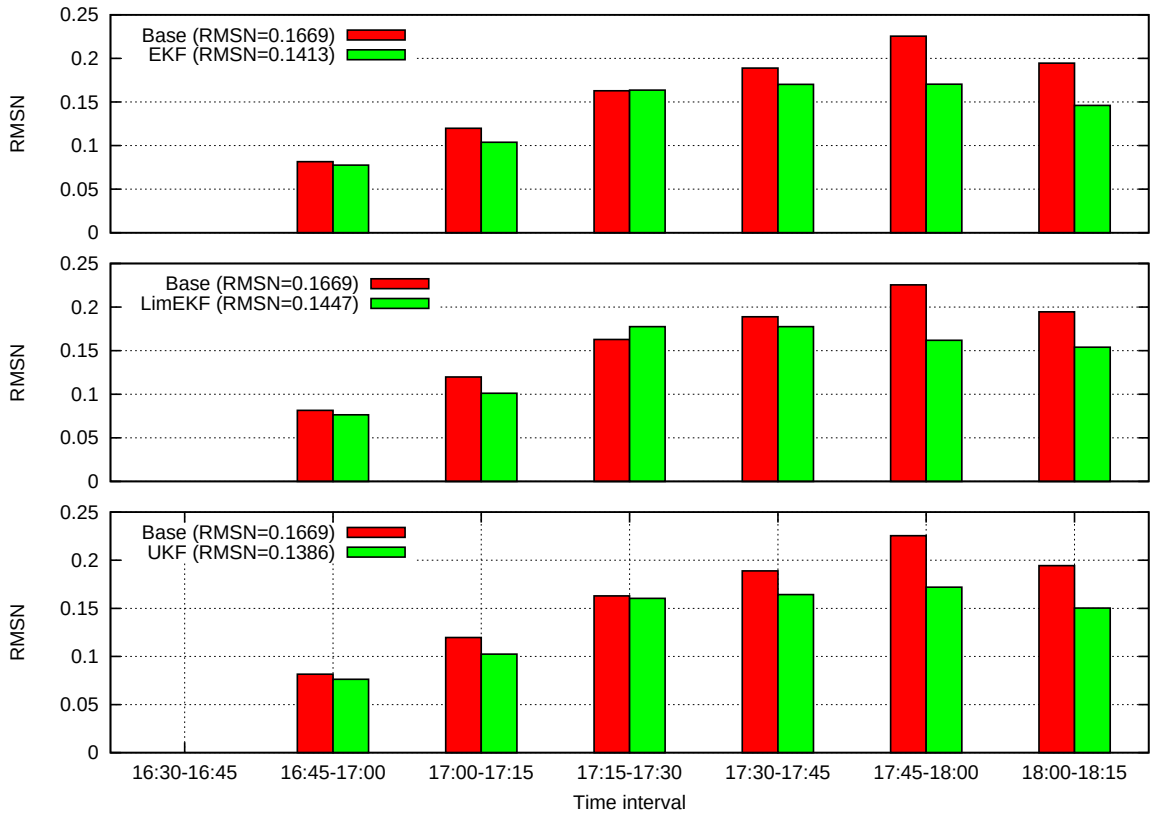

Figure 6-25: Statistics for two–step predicted speeds by interval (wet weather)

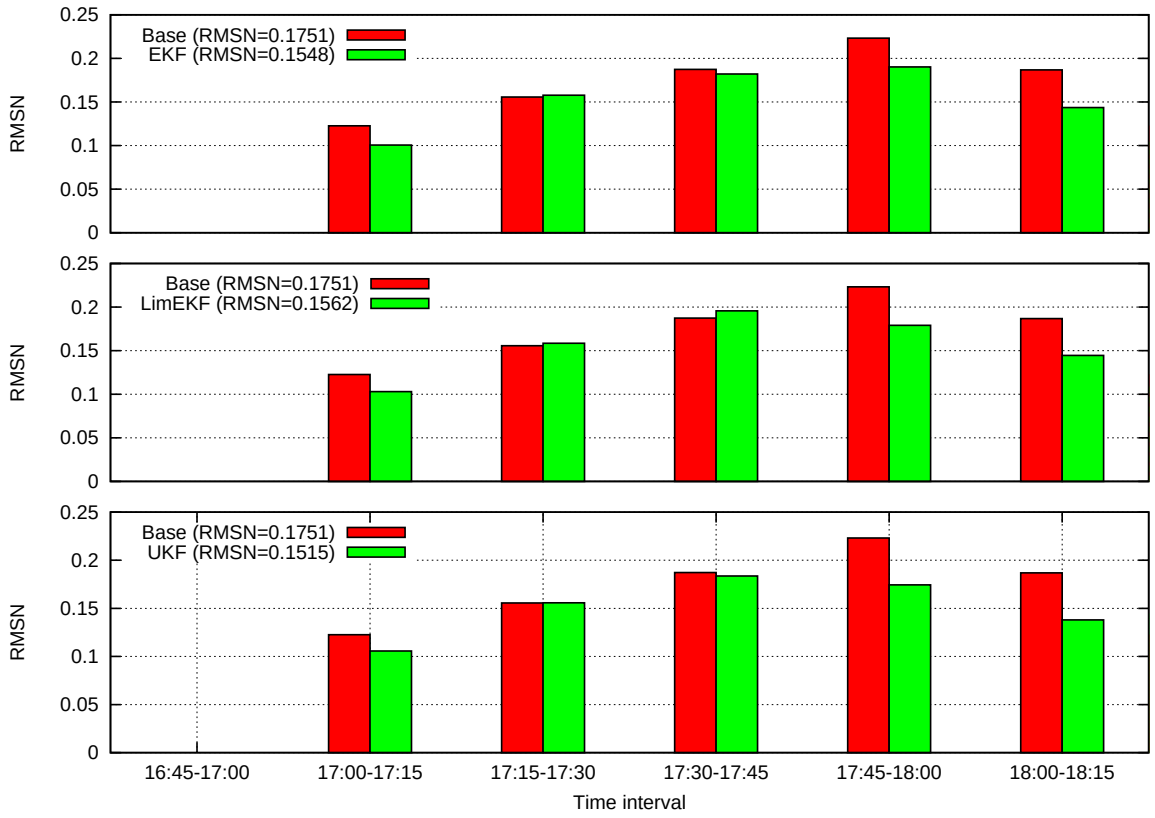

Figure 6-26: Statistics for three–step predicted speeds by interval (wet weather)

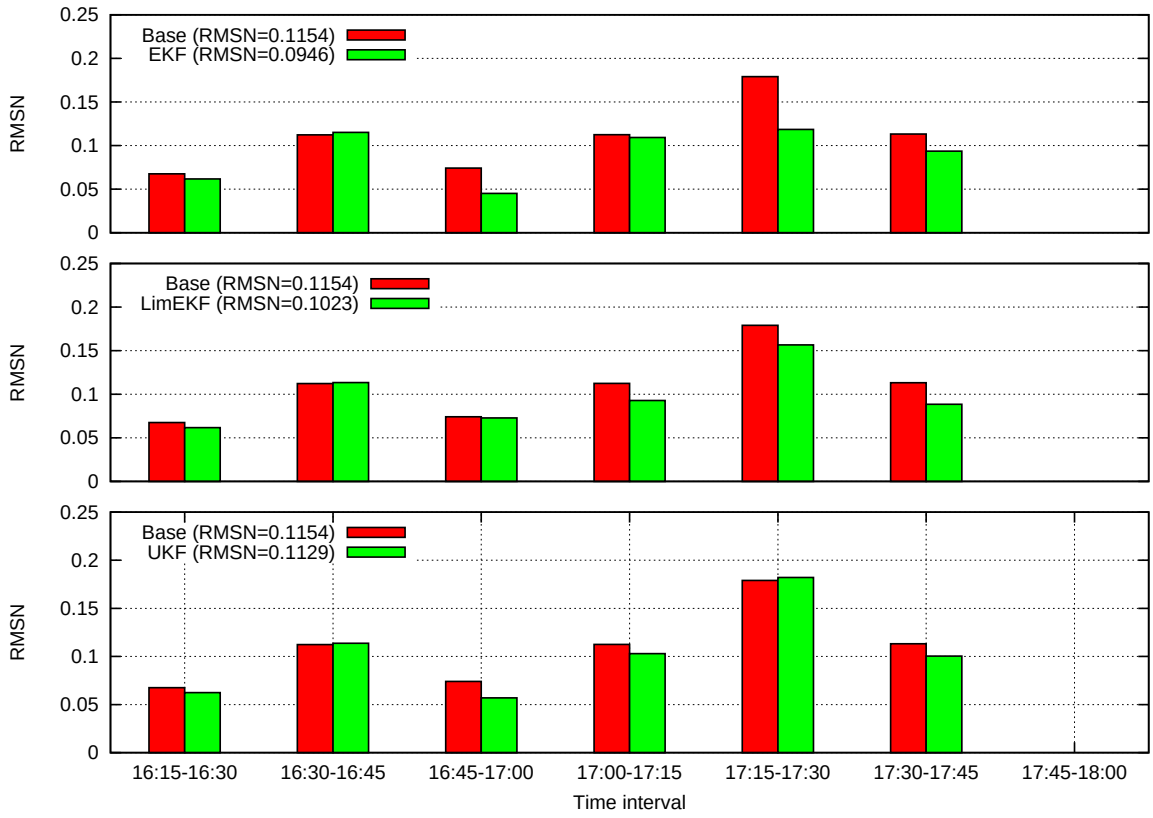

Figure 6-27: Statistics for estimated counts by interval (wet weather)

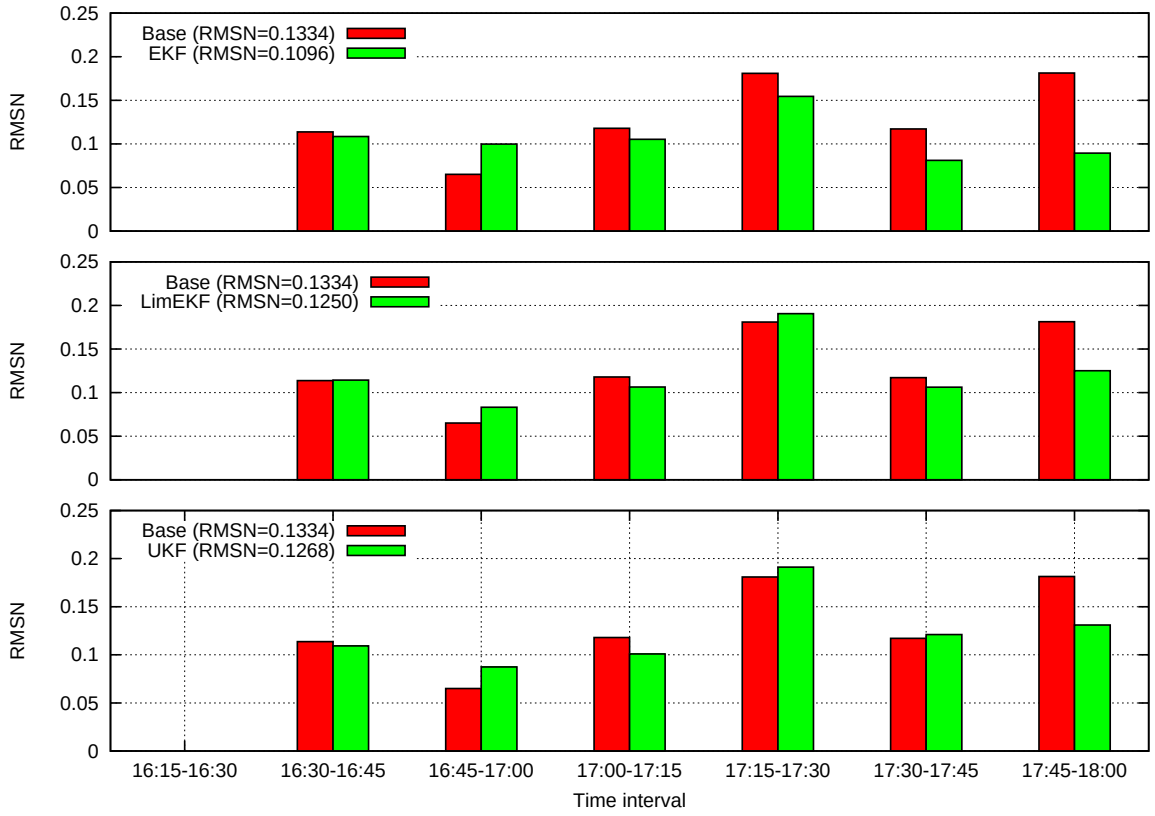

Figure 6-28: Statistics for one–step predicted counts by interval (wet weather)

of the computational complexity. Therefore, the results from this experiment seem to reinforce the conclusion that this algorithm provides an excellent combination of accuracy and computational complexity.

Finally, the performance of the UKF for the counts is similar to that observed for dry weather. In particular, while the algorithm provides similar overall performance as the base algorithm, it seems to provide better performance for some intervals, but worse for other intervals. This is an undesirable property of the algorithm.

The impact of the joint on–line estimation of demand and supply parameters has been demonstrated so far. In the next section we will present the impact of the on–line calibration approach on the estimated parameters.

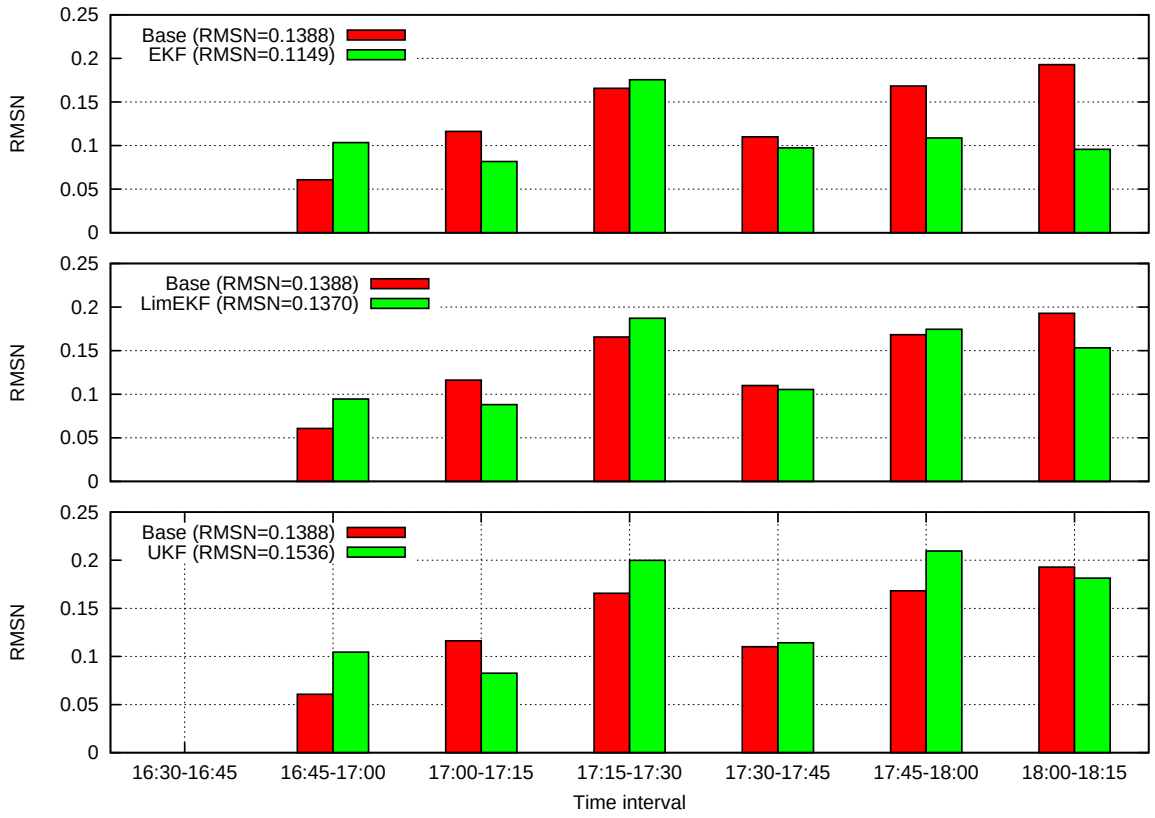

Figure 6-29: Statistics for two–step predicted counts by interval (wet weather)

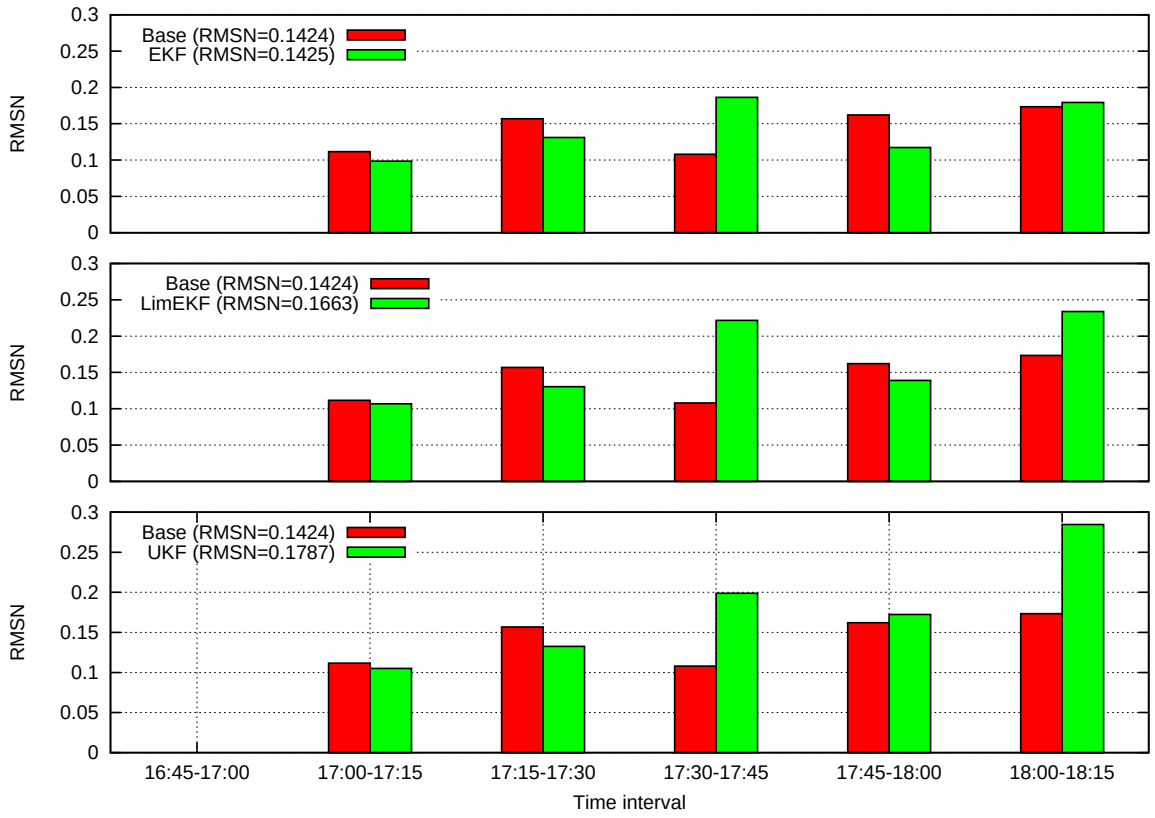

Figure 6-30: Statistics for three–step predicted counts by interval (wet weather)

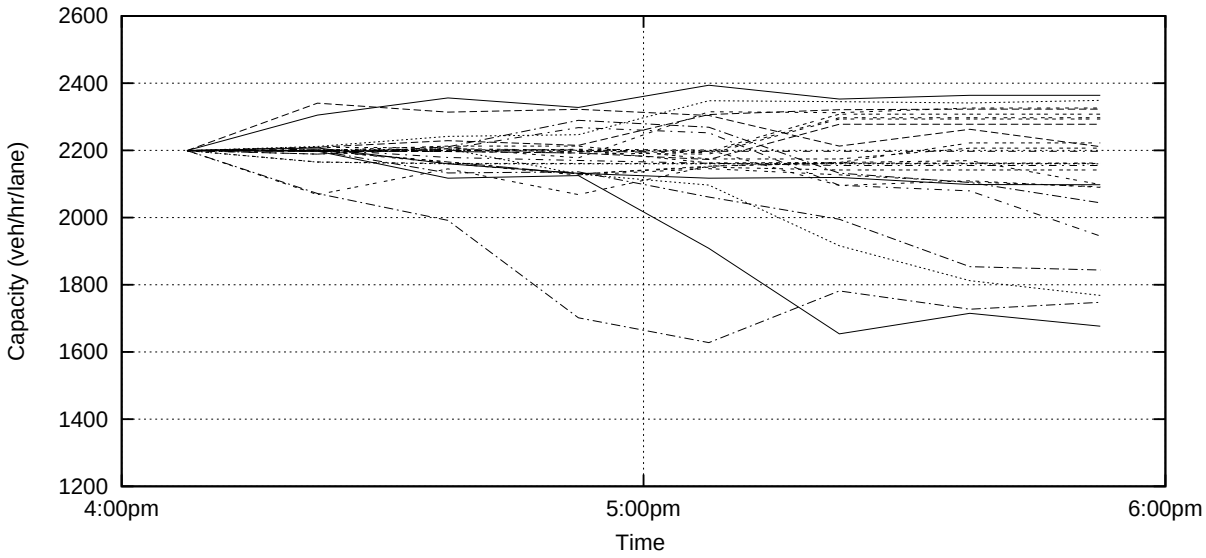

Figure 6-31: Estimated mainline capacities (EKF, dry weather)

6.3.3 Impact on parameters

The on–line calibrated parameters using the EKF algorithm for dry weather are shown in this section.

Figure 6-31 shows the estimated capacities for all mainline segments over time. The capacity in several segments has been increased (from the original 2200 vehicles per hour per lane), sometimes close to 2400 vehicles per hour per lane. The capacity for a few segments, which include weaving and/or merging is reduced to less than 1800 vehicles per hour per lane.

Figure 6-32 shows the estimated capacities for the ramp segments over time. Ramp capacity is generally stable, with the exception of three ramps, in which small decreases (of less than 125 vehicles per hour per lane) are observed. In these ramps the flow does not exceed the estimated capacity.

Figures 6-33 and 6-34 show the evolution of the speed–density relationship parameters over time. The changing values of these parameters result in time–dependent speed–density relationships that more effectively match the observed data.

The free flow speed for the mainline segments follows a trend similar to that of the measured speed, decreasing slightly as traffic increases and recovering towards the end of the peak period. The α and β parameters for the two groups of mainline segments (groups 1 and

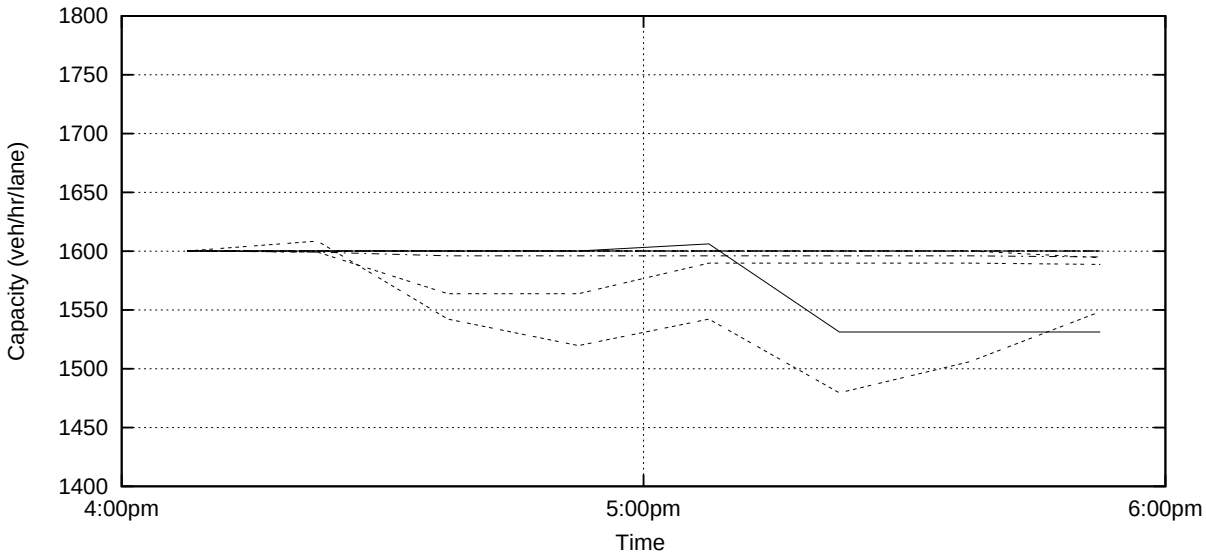

Figure 6-32: Estimated ramp capacities (EKF, dry weather)

2) show similar trends, but with different magnitudes. In particular, the α parameter for both groups decreases during the peak, while the β parameter for both groups increases. A larger decrease (respectively increase) is observed for the second group.

The free–flow speed for the ramp segments shows a small increase from the off–line calibrated value. Similarly, the α parameter increases during the peak period. Finally, the β parameter shows a trend similar to that observed for the mainline segments, namely an increase during the peak period.

The evolution of the estimated "jam density" is shown in Figure 6-34. There is a general trend to increase the jam density, with values reaching 260 vehicles per hour per lane. These values are rather high. However, it should be noted that this should not be seen as an exact representation of the maximum density of the road, but instead as a parameter in the traffic dynamics model.

Finally, it should be noted that the fifth parameter, the minimum density (under which free–flow conditions prevail), does not vary significantly. Note that in the application presented in Section 4.6, where segments were not grouped, two of the parameters maintained their prior values (the minimum and jam density). This observation supports the hypothesis that pooling data from more segments and estimating a single speed–density relationship can overcome potential observability issues (due to the small number of available measurements

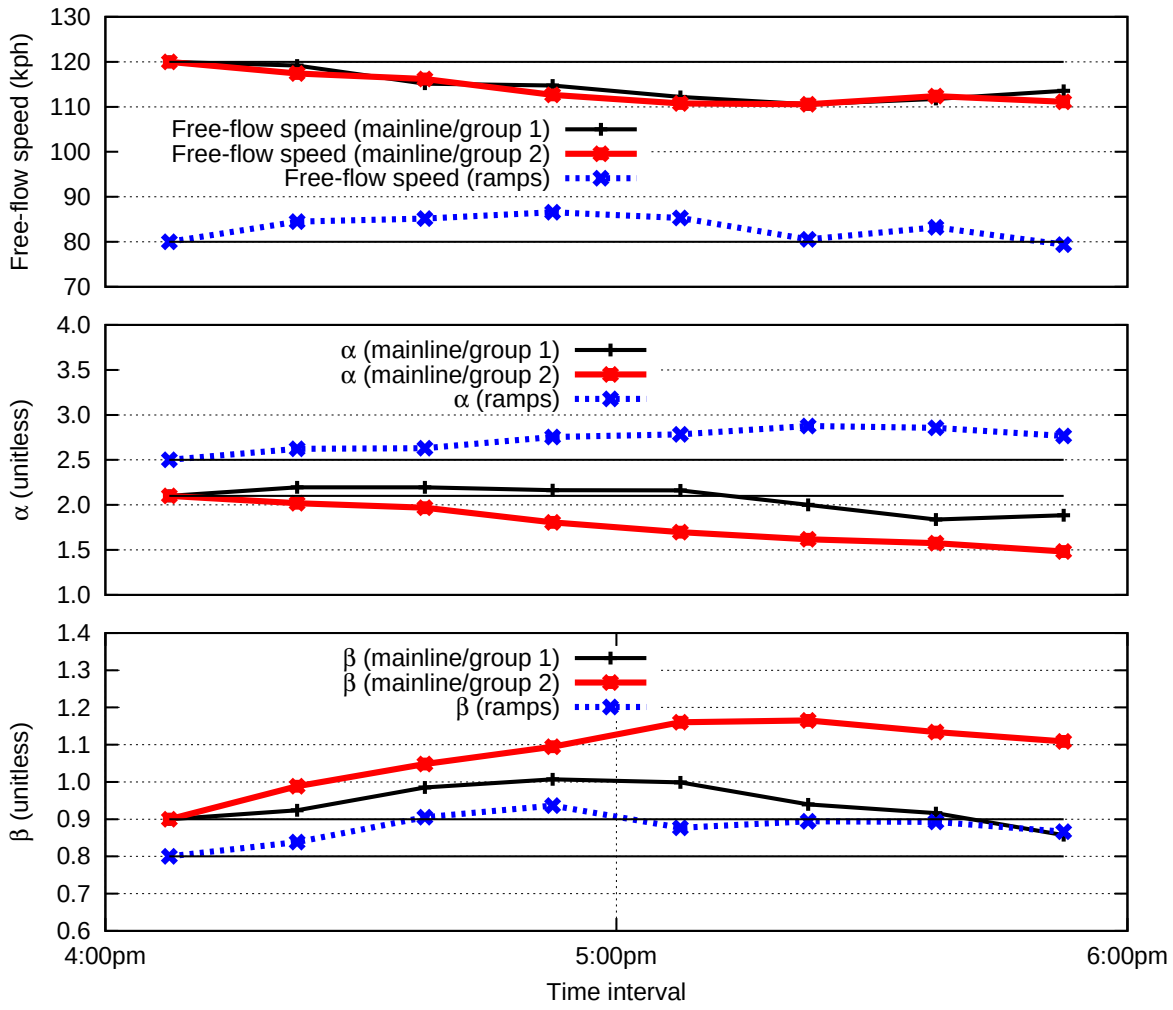

Figure 6-33: Estimated speed–density relationship parameters (EKF, dry weather)

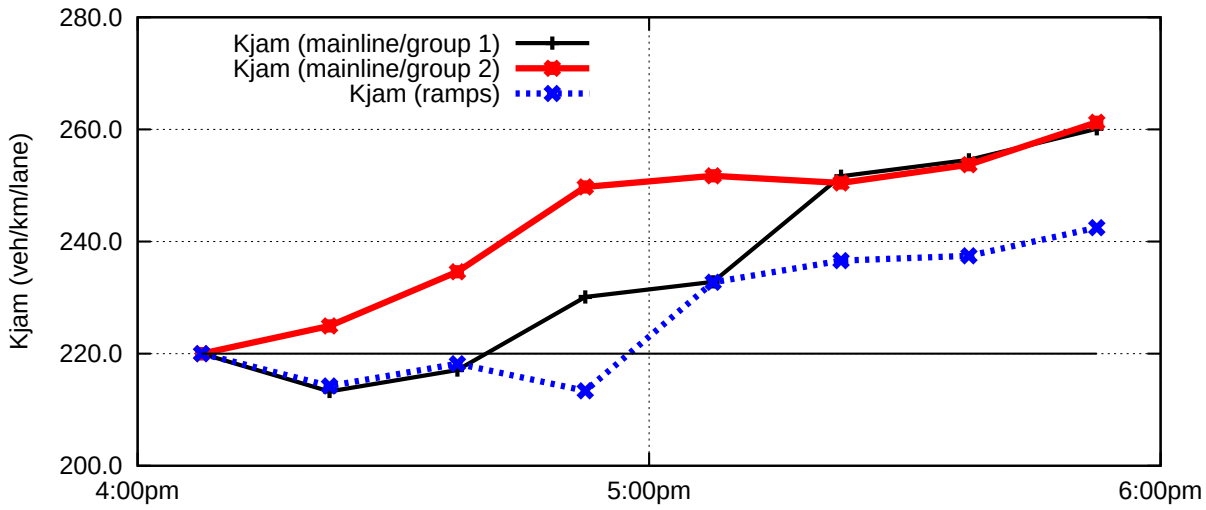

Figure 6-34: Estimated speed–density relationship parameters: Kjam (EKF, dry weather)

for the estimation of the model parameters).

Figure 6-35 shows the estimated capacities for the mainline segments under wet weather over time. A pattern similar to that obtained under dry weather is observed. In particular, the capacity in several segments has been increased (from the original 2200 vehicles per hour per lane) to sometimes close to 2350 vehicles per hour per lane. The capacity for a few segments which include weaving and/or merging is reduced to approximately 1900 vehicles per hour per lane. Again, the flow in these segments does not exceed the estimated capacity.

Figure 6-36 shows the estimated capacities for the ramp segments for wet weather over time. Ramp capacity is generally stable, with the exception of one ramp, in which a small decrease (of less than 50 vehicles per hour per lane) is observed.

Figures 6-37 and 6-38 show the evolution of the speed–density relationship parameters for the wet day over time. The free flow speed for the mainline segments decreases slightly as traffic increases and recovers towards the end of the peak period. The α exponent for group 1 shows very little variability, while the β parameter shows a moderate increase. A similar increase in the value of β is observed for group 2. However, for group 2 the estimated value of the α parameter shows a moderate reduction.

The speed for the ramp segments shows small variability around the off–line calibrated

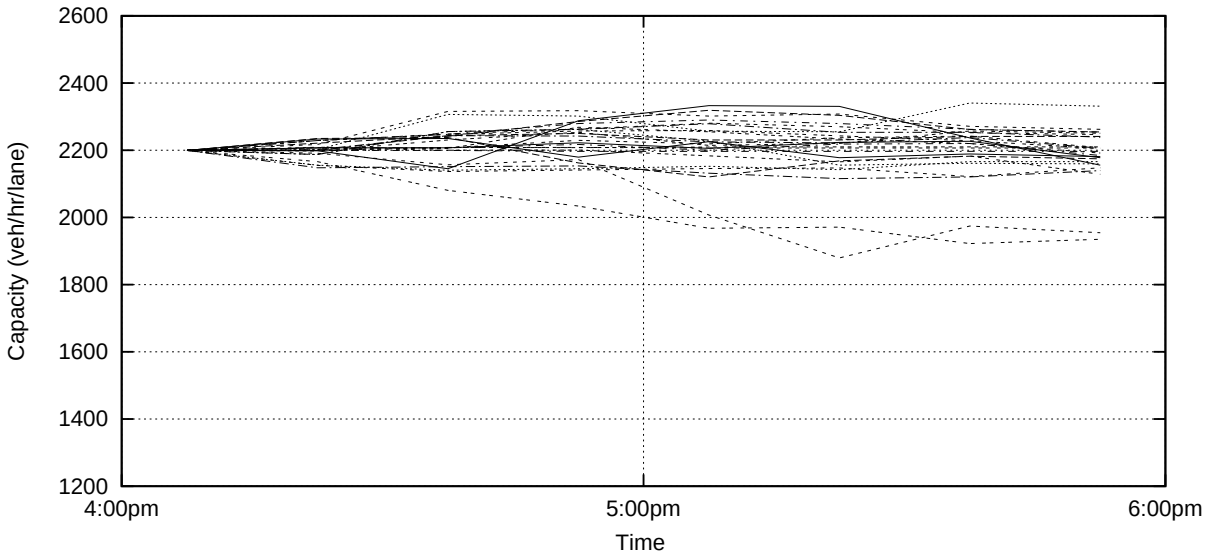

Figure 6-35: Estimated mainline capacities (EKF, wet weather)

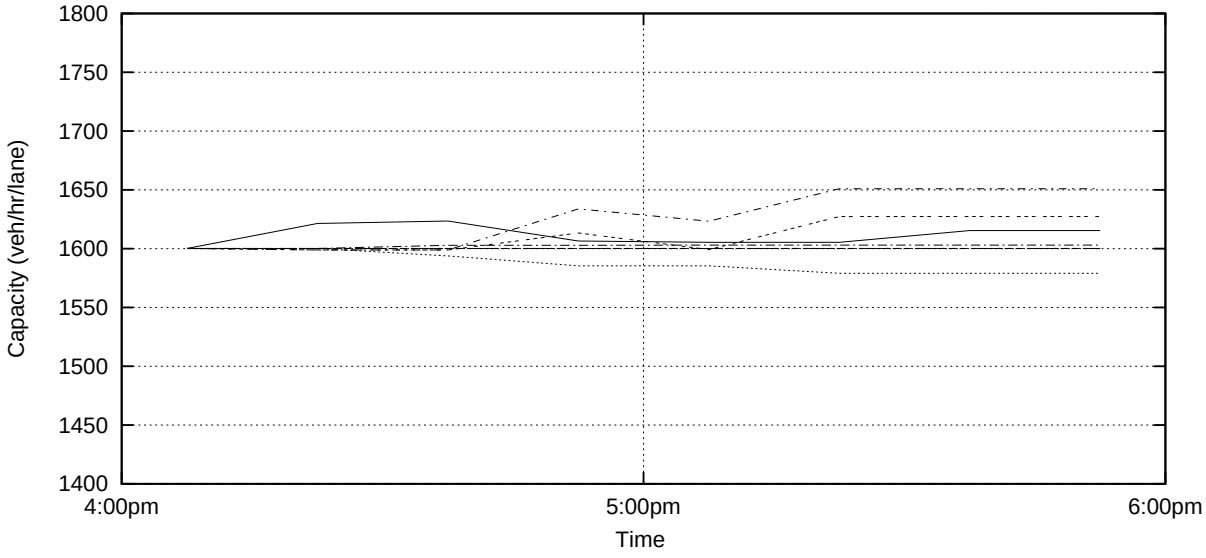

Figure 6-36: Estimated ramp capacities (EKF, wet weather)

value. The α and β parameters show opposite trends, with α decreasing and β increasing as we move into the peak period.

The evolution of the estimated "jam density" is shown in Figure 6-38. Unlike what was experienced during wet weather, there is no general trend to increase the jam density. While the jam density for the second group of mainline segments indeed increases to more than 260 vehicles per hour per lane, moderate decreases are observed for the other two segment groups. The values for the second group are rather high. Again, it should be noted that this should not be seen as an exact representation of the maximum density of the road, but instead as a parameter in the traffic dynamics model.

Finally, it should be noted that the minimum density (under which free–flow conditions prevail) does not vary significantly (similarly to dry weather).

Figure 6-39 graphically illustrates the estimated speed–density relationship for mainline sensors for the 5:30pm–5:45pm time interval for different weather conditions. Under dry weather the speed–density relationship is shifted up and under wet weather the speed–density relationship is shifted down. As expected, at the same density, drivers reduce their speed under wet conditions. This is an interesting demonstration of the ability of the on–line calibration to capture the evolution of the traffic dynamics models.

We now turn our attention to the demand side. Figure 6-40 shows the estimated flows for the three largest OD pairs. Once again, the flows vary smoothly from interval to interval. Furthermore, the trends captured by the two days are similar. This is a reasonable finding, since –except for the weather conditions– the two days are similar and there is no systematic reason for differences in demand.

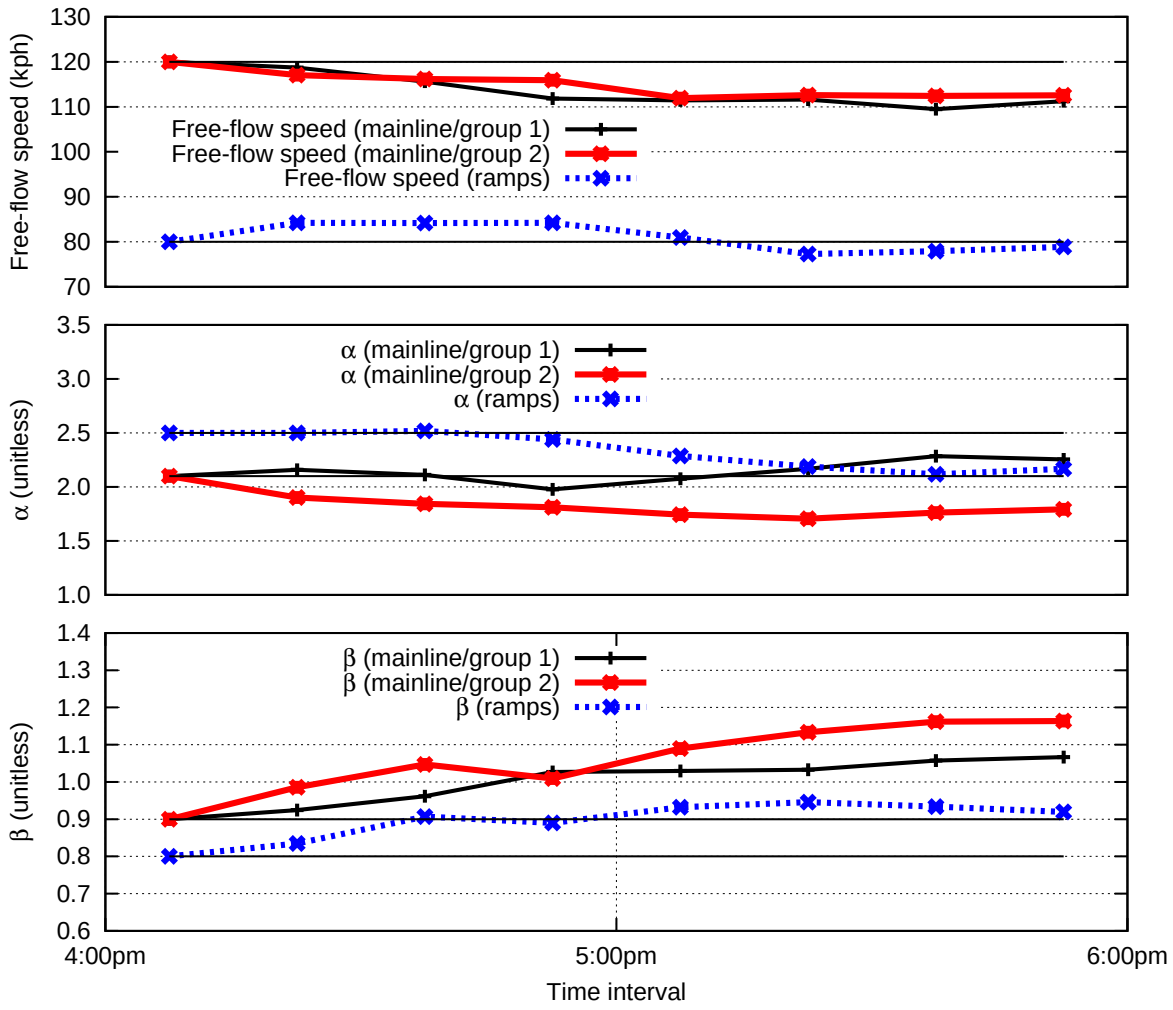

Figure 6-37: Estimated speed–density relationship parameters (EKF, wet weather)

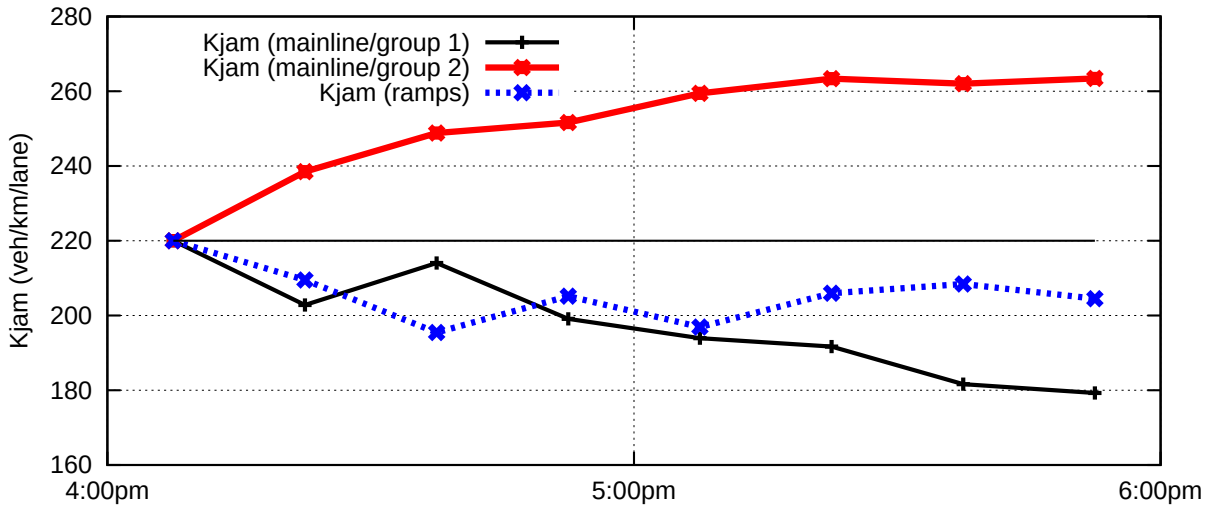

Figure 6-38: Estimated speed–density relationship parameters: Kjam (EKF, wet weather)

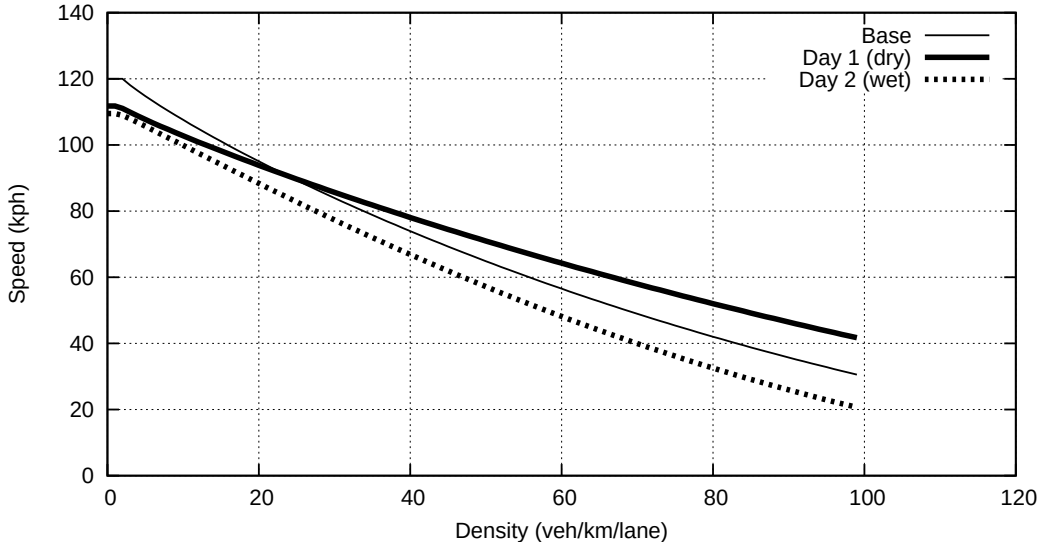

Figure 6-39: Estimated speed–density relationships for mainline sensors (17:15–17:30, EKF)

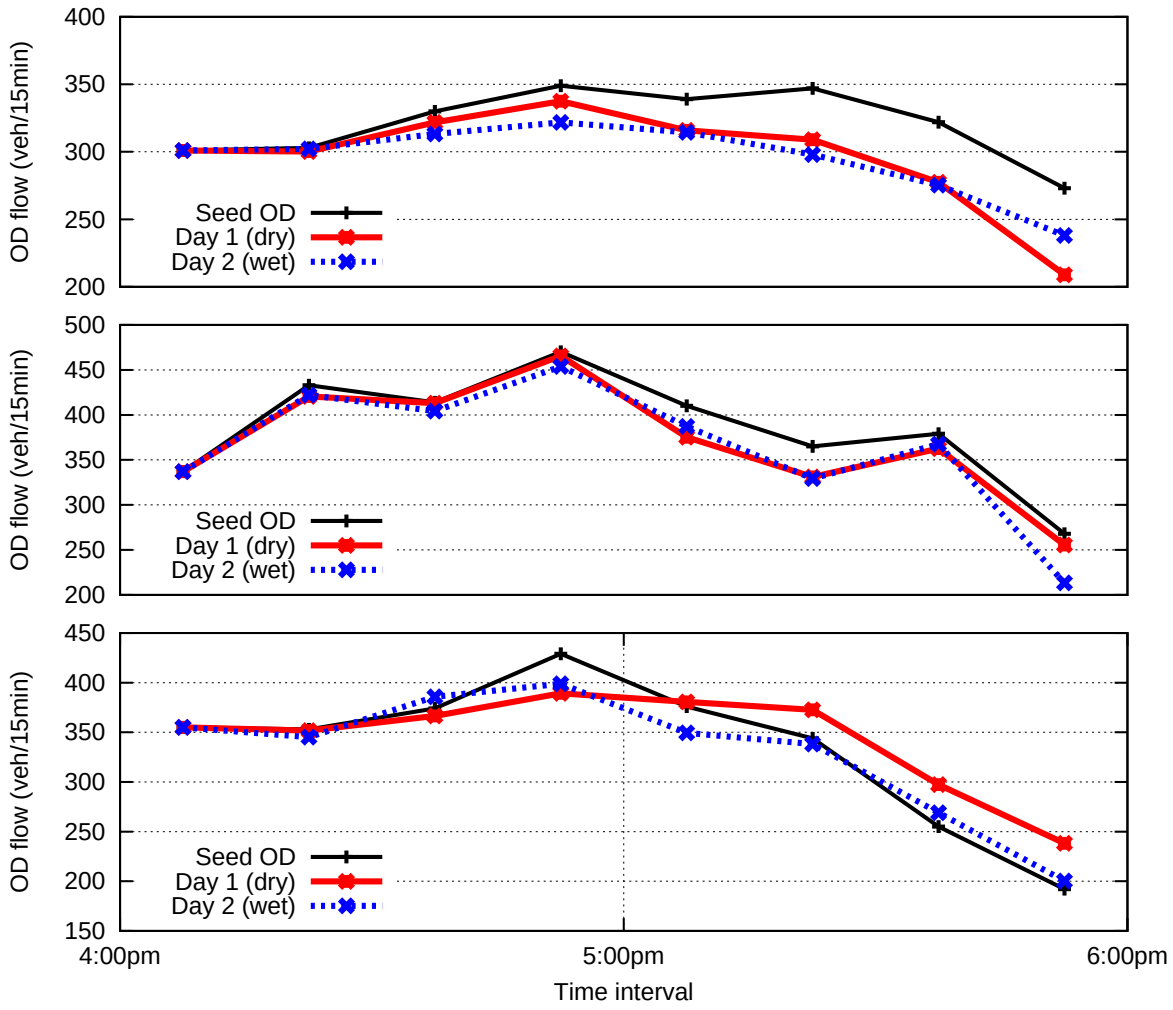

Figure 6-40: Estimated OD flows (three largest OD pairs, EKF)

6.4 Additional analysis

The results presented so far in this chapter provide useful insight into the performance of the on–line calibration approach. However, two questions arise:

- As expected, the performance of the on–line calibration approach decreases as the prediction horizon increases. However, in this case, the on–line calibration approach provided a worse performance (in prediction of counts) than the base case. This could be a symptom of over–fitting, which (as discussed earlier) is a natural concern in parameter estimation.
- The algorithm seems to be capturing predicted speeds much better that it is capturing predicted counts.

In this section, we present the results of some further experimentation aimed at providing additional insight on these two questions.

As discussed before, minute–by–minute speed measurements have been used as inputs, while count measurements have been aggregated to fifteen minute intervals. The decision to use fifteen–minute counts is based on the requirements of the OD estimation approach, used as the base case (which assumes that the counts are aggregated in intervals equal to the estimation/prediction interval).

As a result, a larger number of speed measurements (fifteen per sensor per interval) have been used, as opposed to a single count measurement per sensor per interval. As a consequence, the speed measurements have a higher weight on the algorithm. In order to correct for this, the variance–covariance matrices have been adjusted to reflect these observations. In order to increase the weight on the count measurements, the variance of these measurements has been reduced by a factor of ten.

Furthermore, in order to investigate the over–fitting concern, the weights of the *a priori* values of the parameters have also been modified. In particular, the weights have been doubled (by dividing the corresponding covariances by two), effectively implying that we are more confident about the *a priori* values of the parameters. Furthermore, a longer prediction horizon has been used (five–step prediction, corresponding to prediction 75 minutes into the future).

Table 6.6: Summary results (additional runs, dry day)

	Estimated		One–step predicted		Two–step predicted	
Algorithm	speeds	counts	speds	counts	speeds	counts
Base	0.1266	0.1288	0.1494	0.1540	0.1441	0.1666
LimEKF	0.1121	0.1092	0.1249	0.1325	0.1346	0.1698
% improv	11.5%	15.2%	16.4%	14.0%	6.6%	−1.9%

	Three–step predicted		Four–step predicted		Five–step predicted	
Algorithm	speeds	counts	speds	counts	speeds	counts
Base	0.1604	0.1904	0.1660	0.2378	0.1772	0.2878
LimEKF	0.1505	0.1765	0.1537	0.2322	0.1556	0.2858
% improv	6.2%	7.3%	7.4%	2.4%	12.2%	0.7%

The Limiting EKF was selected as the most appropriate algorithm for this analysis, as this algorithm has been shown to be the most practical (from a computational point of view) and therefore the most likely to be used in the field. Since the Limiting EKF requires a pre–computed Kalman gain, the EKF algorithm was run first (with the updated variance/covariance matrices as inputs) to generate that. The Kalman gain for the Limiting EKF was computed as the average of the Kalman gain matrices generated by the EKF.

The estimation and prediction results for the LimEKF (for both speeds and counts) are presented in Table 6.6 and Figures 6-41 (speeds) and 6-42 (counts). The results provide a more balanced performance in the prediction of counts and speeds, with benefits (or performance equivalent to the base case) even for five–step prediction. This finding confirms the hypothesis that the use of minute–by–minute speeds (as opposed to fifteen–minute counts) results in additional weight to be given to the speeds. This issue can be overcome (as demonstrated) by decreasing the variance (and therefore increasing the weight) of the count measurements.

A second key observation is that by increasing the weight of the *a priori* values of the parameters, it is possible to avoid over–fitting. The benefits of the on–line calibration

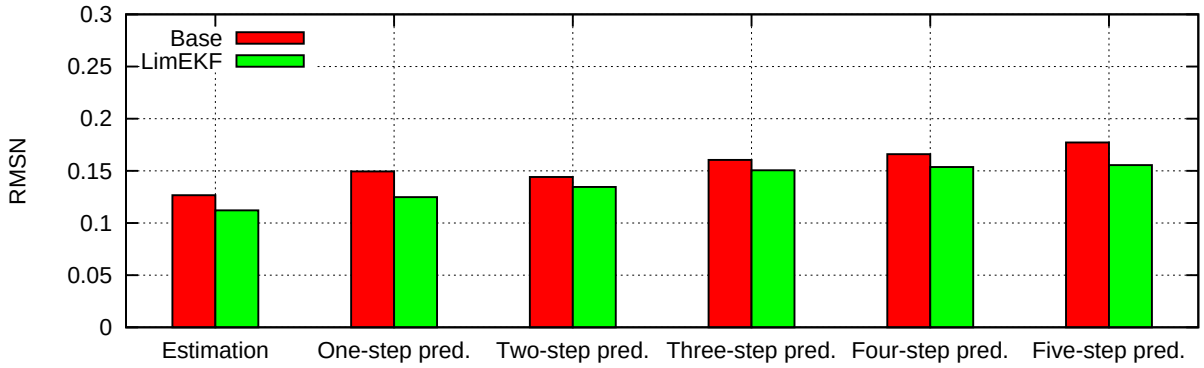

Figure 6-41: Summary results with modified weights (speeds, dry weather, LimEKF)

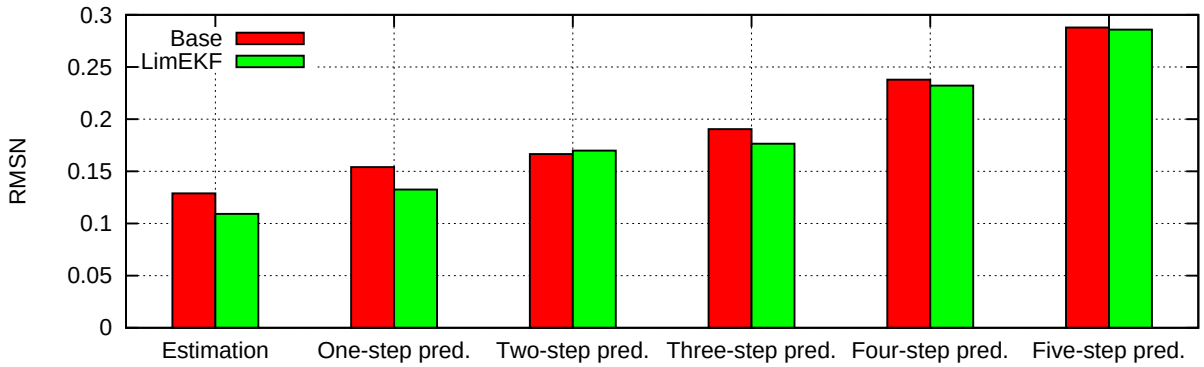

Figure 6-42: Summary results with modified weights (counts, dry weather, LimEKF)

decrease as the prediction horizon increases and they converge to the performance of the base case. This finding is confirming that the (estimated) supply parameters are now closer to their *a priori* values. Therefore, the adjustment to the variances is having the desirable effect.

6.5 Major findings

We conclude this chapter with a summary of the major findings from the case study.

- Joint on–line calibration of demand and supply parameters can improve estimation and prediction accuracy of a DTA system. In the case study, benefits between 10% and 20% were obtained (over the case when only OD flows were estimated on–line). While the results obtained from this real network application are promising, they should be validated in further empirical studies. In particular, the scalability of the

approach to larger, more complex networks needs to be investigated.

- The on–line calibration approach is computationally feasible. Even when using the "exact" algorithms (EKF and UKF) the model achieves "real-time" performance in a small, but non-trivial real network. Furthermore, the Limiting EKF provides accuracy comparable to that of the *best* algorithm (EKF), but with computational complexity which is order(s) of magnitude lower than the other algorithms.

- The EKF algorithm has more desirable properties than the UKF algorithm (at least in this application)

  - The EKF generally outperforms UKF. It should be noted, however, that the full power of the UKF algorithm is not exhibited in this application, because the transition equation is already linear (and therefore, there was no need to linearize it, thus introducing an additional approximation in the EKF solution)

  - The EKF and UKF algorithms have similar computational properties, with $2n$ function evaluations (and hence runs of the simulator) required for each run of the EKF and $2n + 1$ function evaluations required for the UKF (where n is the dimension of the state vector).

- The Limiting EKF provides accuracy comparable to that of the *best* algorithm (EKF), but with computational complexity which is order(s) of magnitude lower than the other algorithms. One interesting property of the LimEKF algorithm is that it requires a single function evaluation irrespective of the dimension of the state vector (while the computational complexity of the EKF and UKF algorithms increases proportionally with the state dimension). Therefore, the computational benefits increase as the dimension of the problem increases.

- The approach appears to be robust and can be applied to different conditions. However, further investigation of the robustness of the approach is required.

Chapter 7

Conclusion

Contents

7.1 Summary and findings

An on–line calibration approach for dynamic traffic assignment systems has been developed. The approach is general and flexible and makes no assumptions on the type of the DTA system, the models or the data that it can handle. Therefore, it is applicable to a wide variety of tools including simulation–based and analytical, as well as microscopic and macroscopic models.

The objective of the on–line calibration approach is to introduce a systematic procedure that will use the available data to *steer* the model parameters to values closer to the *realized* ones. The output of the on–line calibration is therefore a set of parameter values that —when used as input for traffic estimation and prediction— minimizes the discrepancy between the simulated (estimated and predicted) and the observed traffic conditions. The scope of the on–line calibration is neither to duplicate nor to substitute for the off–line calibration process. Instead, the two processes are complementary and synergistic in nature.

A classical technique for dealing with dynamic systems is *state–space* modeling. The on–line calibration approach is formulated as a state–space model, comprising transition equations that capture the evolution of the state vector over time, and measurement equations that capture the mapping of the state vector on the measurements. *A priori* values of the model parameters provide *direct* measurements of the unknown parameters. Surveillance information, on the other hand, can be used to formulate *indirect* measurement equations, where the output of the simulator model (when the unknown set of parameter values is used as input) would match the surveillance information. The state vector is defined in terms of deviations of the parameters and inputs that need to be calibrated from available estimates. This reformulation incorporates structural information from previous intervals into the formulation.

State–space models have been extensively studied and efficient algorithms have been developed, such as the Kalman Filter for linear models. However, the on–line calibration formulation is not linear (due to the indirect measurement equation). Therefore, modified Kalman Filter methodologies have been presented. The most straightforward extension is the Extended Kalman Filter (EKF), in which optimal quantities are approximated via first order Taylor series expansion (linearization) of the appropriate equations (Kalman, 1960;

Gelb, 1974).

The Limiting EKF is a variation of the EKF that eliminates the need to perform the most computationally intensive steps of the algorithm on–line. The use of the Limiting EKF provides dramatic improvements in terms of computational performance.

The Unscented Kalman Filter (UKF) is an alternative filter that uses a deterministic sampling approach (Julier et al., 1995; Julier and Uhlmann, 1997; Wan et al., 2000; Wan and van der Merwe, 2000; van der Merwe et al., 2000). The computational complexity of the UKF is of the same order as that of the EKF.

Empirical results suggest that joint on–line calibration of demand and supply parameters can improve estimation and prediction accuracy of a DTA system. While the results obtained from this real network application are promising, they should be validated in further empirical studies. In particular, the scalability of the approach to larger, more complex networks needs to be investigated.

The results also suggest that —in this application— the EKF has more desirable properties than the UKF. The Limiting EKF provides accuracy comparable to that of the *best* algorithm (EKF), while providing order(s) of magnitude improvement in computational performance. Furthermore, the LimEKF algorithm is that it requires a single function evaluation irrespective of the dimension of the state vector (while the computational complexity of the EKF and UKF algorithms increases proportionally with the state dimension). This property makes this an attractive algorithm for large–scale applications.

7.2 Research contributions

This research makes several concrete contributions to the state–of–the–art. Specifically:

- A comprehensive framework for the on–line calibration of a Dynamic Traffic Assignment system is developed. The on–line calibration approach is expressed in a compact form with the following features:
  - Integrates the OD estimation and supply parameter estimation problems into a single formulation. Demand and supply parameters are thus estimated jointly and simultaneously, internalizing demand–supply interactions.

  - Is generic and applicable to any DTA system. In particular, the on–line calibration approach does not make any assumptions on the specific models that comprise the DTA system. Therefore, it is applicable to systems with very different characteristics (e.g. analytical versus simulation–based, or microscopic versus macroscopic).
  - Has flexible data requirements. The approach can easily incorporate all available surveillance information —for example Automated Vehicle Identification (AVI) or probe vehicle data.

- New solution algorithms are applied to the problem. Besides the Extended Kalman Filter (EKF), which has been a well–established algorithm for dealing with non–linear state–space models, the Limiting EKF and the Unscented Kalman Filter (UKF) are considered.
  - The accuracy achieved by the Limiting EKF is comparable to the EKF, but the computational cost is order(s) of magnitudes lower. The Limiting EKF makes the on–line calibration approach computationally feasible.

7.3 Directions for further research

Methodological, algorithmic and application–related directions for further research are outlined in this section.

- **Behavioral model parameters** As mentioned in Section 2.1.2 that —while driving behaviors do change over time— the underlying behavioral parameters do not change rapidly (but as a result of longer–term learning processes and therefore they should not be captured in the on–line calibration). However, it can be argued that variations in the *effective* behavioral parameters can be observed due to other reasons, such as variations in the traffic mix. For example, a major special event (such as a sporting event or a concert) may result in large numbers of drivers approaching or leaving a location at the same time. In that case, the change in the composition of the traffic could result in behaviors significantly different from the habitual.

  The ability of a traffic estimation and prediction system to detect (and react to) such

phenomena would be a desirable property. The flexibility and generality of the on–line calibration approach allows the incorporation of additional parameters (such as the behavioral model parameters) into the formulation.

- **Dynamic segment groups** A common approach to dealing with speed–density relationships involves the grouping of segments with similar characteristics into homogeneous groups and the estimation of a single relationship for each group. The use of fixed classes of segments, however, does have drawbacks. For example, if an incident occurs in one segment (thus significantly altering the traffic dynamics for this segment), the model will not be able to distinguish between the properties of this segment and the properties of the other segments in the group. Within the on–line calibration approach, it is possible to estimate speed–density relationships per individual segment or per group. Moving from one option to the other, however, is not straightforward. An approach that would allow for the dynamic grouping of segments could provide more flexibility in this respect.

- **Further experimental analysis** The on–line calibration approach has been successfully demonstrated on one real, if simple, network. These findings, however, can not be generalized; further applications in different networks need to be performed. In particular, the scalability of the model to larger networks, as well as its robustness when applied to more complex network configurations need to be demonstrated.

- **Different algorithms** Among the algorithms that were presented in this thesis, the UKF produced the least favorable results. Further analysis is required into the reasons that the UKF did not perform as well. It is possible, for example, that the number of sigma points used by the UKF is not sufficient to accurately capture the first moments of the random variables (given the large state dimension). Other algorithms of the same "family" could be considered, as they may provide better properties and be more suitable in this context. Examples of such algorithms include the Scaled Unscented Kalman Filter (Julier, and J. K. Uhlmann, 2002) and Particle Filters. Particle filters (Gordon, 2003; Arulampalam et al., 2002; Pitt and Shephard, 1999) are sequential Monte–Carlo methods based upon point mass (or "particle") representations of probability densities, which can be applied to any state–space model, and which generalize the traditional Kalman filtering methods.

Furthermore, the applicability of other solution algorithms, such as the generalization of secant methods for the solution of systems of nonlinear equations developed by Crittin and Bierlaire (2003), can be investigated.

Bibliography

Antoniou, C. (1997). Demand simulation for dynamic traffic assignment. Master's thesis, Massachusetts Institute of Technology.

Antoniou, C., Ben-Akiva, M., Bierlaire, M., and Mishalani, R. (1997). Demand simulation for dynamic traffic assignment. In *Proceedings of the 8th IFAC Symposium on Transportation Systems*, Chania, Greece.

Antoniou, C., Ben-Akiva, M., and Koutsopoulos, H. N. (2004). Incorporating automated vehicle identification into origin–destination estimation. *Accepted for publication in the Transportation Research Record*.

Arulampalam, S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188.

Ashok, K. (1996). *Estimation and Prediction of time-dependent Origin-Destination Flows*. PhD thesis, Massachusetts Institute of Technology.

Ashok, K. and Ben-Akiva, M. (1993). Dynamic O-D matrix estimation and prediction for real-time traffic management systems. In Daganzo, C., editor, *Transportation and Traffic Theory*, pages 465–484. Elsevier Science Publishing.

Ashok, K. and Ben-Akiva, M. (2000). Alternative approaches for real-time estimation and prediction of time-dependent origin–destination flows. *Transportation Science*, 34(1):21–36.

Ashok, K. and Ben-Akiva, M. (2002). 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.

Balakrishna, R. (2002). Calibration of the demand simulator in a dynamic traffic assignment system. Master's thesis, Massachusetts Institute of Technology.

Balakrishna, R., Koutsopoulos, H. N., and Ben-Akiva, M. (2004). Evaluation of the estimation and prediction capabilities of a dynamic traffic assignment system. *Submitted to the 16th International Symposium on Transportation and Traffic Theory, University of Maryland, College Park*.

Ben-Akiva, M., Bierlaire, M., Koutsopoulos, H. N., and Mishalani, R. (2002). Real-time simulation of traffic demand-supply interactions within DynaMIT. In Gendreau, M. and Marcotte, P., editors, *Transportation and network analysis: current trends*, pages 19–36. Kluwer Academic Publishers. Miscellenea in honor of Michael Florian.

Ben-Akiva, M., DePalma, A., and Kaysi, I. (1991). Dynamic network models and driver information systems. *Transportation Research A*, 25(5).

Bierlaire, M. and Crittin, F. (2004). An efficient algorithm for real–time estimation and prediction of dynamic od tables. *Operations Research*, 52(1).

Bottom, J. A. (2000). *Consistent Anticipatory Route Guidance*. PhD thesis, Massachusetts Institute of Technology.

Cascetta, E., Inaudi, D., and Marquis, G. (1993). Dynamic estimators of origin-destination matrices using traffic counts. *Transportation Science*, 27(4):363–373.

Chui, C. K. and Chen, G. (1999). *Kalman Filtering with Real-Time Applications*. Springer-Verlag.

Crittin, F. (2003). *New algorithmic methods for real–time transportation problems*. PhD thesis, Ecole Polytechnique Federale de Lausanne.

Crittin, F. and Bierlaire, M. (2003). A generalization of secant methods for solving nonlinear systems of equations. In *Proceeding of the 3rd Swiss Transportation Research Conference, Ascona, Switzerland*. http://www.strc.ch/Paper/crttin_6.pdf.

Doan, D. L., Ziliaskopoulos, A., and Mahmassani, H. (1999). On-line monitoring system for real-time traffic management applications. *Transportation Research Record 1678*, pages 142–149.

FHWA (2001). Managing our congested streets and highways. Technical report, Federal Highway Administration, US Department of Transportation, Washington, D.C. Publication No. FHWA-OP-01-018.

Gelb, A., editor (1974). *Applied Optimal Estimation*. M.I.T. Press.

Golub, G. H. and van Loan, C. F. (1996). *Matrix Computations*. Johns Hopkins Series in the Mathematical Sciences. Johns Hopkins University Press, 3rd edition edition.

Gordon, N. J. (2003). Beyond the Kalman Filter: Particle filters for tracking applications. In *Workshop on New Directions in Signal Processing in the XXI century. Lake Louise, Alberta, Canada*.

Greene, W. H. (2000). *Econometric analysis*. Prentice-Hall, Inc., 4th edition.

Greenshields, B. D. (1935). A study of traffic capacity. In *Highway Research Board Proceedings 14*, pages 448–477.

Hall, F. L. (1997). Traffic stream characteristics. In N. H. Gartner, C. J. M. and Rathi, A. K., editors, *Revised Monograph of Traffic Flow Theory*. Federal Highway Administration. http://www.tfhrc.gov/its/tft/tft.htm.

HCM (2000). *Highway Capacity Manual*. Transportation Research Board, Washington, D.C.

He, R., Miaou, S., Ran, B., and Lan, C. (1999). Developing an on-line calibration process for an analytical dynamic traffic assignment model. In *Presented at the 78th Annual Meeting of the Transportation Research Board, Washington, D.C.*

Hellinga, B. and Fu, L. (2002). Reducing bias in probe-based arterial link travel time estimates. *Transportation Research - Part C*, 10:257–273.

Huynh, N., Mahmassani, H., and Tavana, H. (2002). Adaptive speed estimation using transfer function models for real-time dynamic traffic assignment operation. In *Proceedings of the 81st Annual Meeting of the Transportation Research Board*.

Julier, S. and Uhlmann, J. (1996). A general method for approximating nonlinear transformations of probability distributions. Technical report, Robotics Research Group, Department of Engineering Science, University of Oxford, Oxford, OX1 3PJ United Kingdom.

Julier, S. and Uhlmann, J. (1997). A new extension of the kalman filter to nonlinear systems. In *Proceedings of Aerosense: The 11th International Symposium of Aerospace/Defense Sensing, Simulation and Controls, Orlando, FL.*, volume Multi Sensor Fusion, Tracking and Resource Management II.

Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. (1995). A new approach for filtering nonlinear systems. In *Proceedings of the American Control Conference*, pages 1628–1632.

Julier, and J. K. Uhlmann, S. J. (2002). The Scaled Unscented Transformation. In *Proceedings of the IEEE American Control Conference*, pages 4555–4559, Anchorage AK, USA. IEEE.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering (ASME)*, 82D:35–45.

Kunde, K. (2002). Calibration of mesoscopic traffic simulation models for dynamic traffic assignment. Master's thesis, Massachusetts Institute of Technology.

Mahmassani, H. S. (2001). Dynamic network traffic assignment and simulation methodology for advanced system management applications. *Networks and Spacial Economics*, 1(3):267–292.

Mouskos, K. C., Niver, E., Pignataro, L., Lee, S., Antoniou, N., and Papadopoulos, L. (1998). TRANSMIT system evaluation - final report. Technical report, New Jersey Institute of Technology.

Peeta, S. and Bulusu, S. (1999). Generalized singular value decomposition approach for consistent on–line dynamic traffic assignment. *Transportation Research Record 1667*, pages 77–87.

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446).

Qin, X. and Mahmassani, H. (2004). Adaptive calibration of dynamic speed-density relations for online network traffic estimation and prediction applications. In *Proceedings of the 83rd Annual Meeting of the Transportation Research Board*, Washington, D.C.

Schrank, D. and Lomax, T. (2003). 2003 urban mobility report. Technical report, Texas Transportation Institute, The Texas A&M University System.

Sorenson, H. W., editor (1985). *Kalman Filtering: Theory and Application*. IEEE Press, New York.

Sundaram, S. (2002). Development of a dynamic traffic assignment system for short–term planning applications. Master's thesis, Massachusetts Institute of Technology.

Tavana, H. and Mahmassani, H. (2000). Estimation and application of dynamic speed-density relations by using transfer function models. *Transportation Research Record 1710*, pages 47–57.

University of Maryland (2003). DYNASMART home page. http://www.dynasmart.umd.edu/, Accessed: October 27, 2003.

van Arem, B. and van der Vlist, M. J. M. (1992). An on–line procedure for estimating current capacity. Technical Report INRO-VVG 1991-17, TNO Institute of Spatial Organization, Delft.

van der Merwe, R., Doucet, A., de Freitas, N., and Wan, E. (2000). The Unscented Particle Filter. Technical Report Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department.

Wan, E. and van der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In *Proceedings of IEEE Symposium 2000 (AS-SPCC)*. Lake Louise, Alberta, Canada.

Wan, E., van der Merwe, R., and Nelson, A. T. (2000). Dual estimation and the unscented transformation. In Solla, S. A., Leen, T. K., and Müller, K.-R., editors, *Advances in Neural Information Processing Systems 12*, pages 666–672. MIT Press.

Wan, E. A. and van der Merwe, R. (2001). *Kalman Filtering and Neural Networks*, chapter The Unscented Kalman Filter. John Wiley and Sons.

Wang, Y. and Papageorgiou, M. (2004). Real-time Freeway Traffic State Estimation based on Extended Kalman Filter: A General Approach. *Accepted for publication in Transportation Research Part B*.