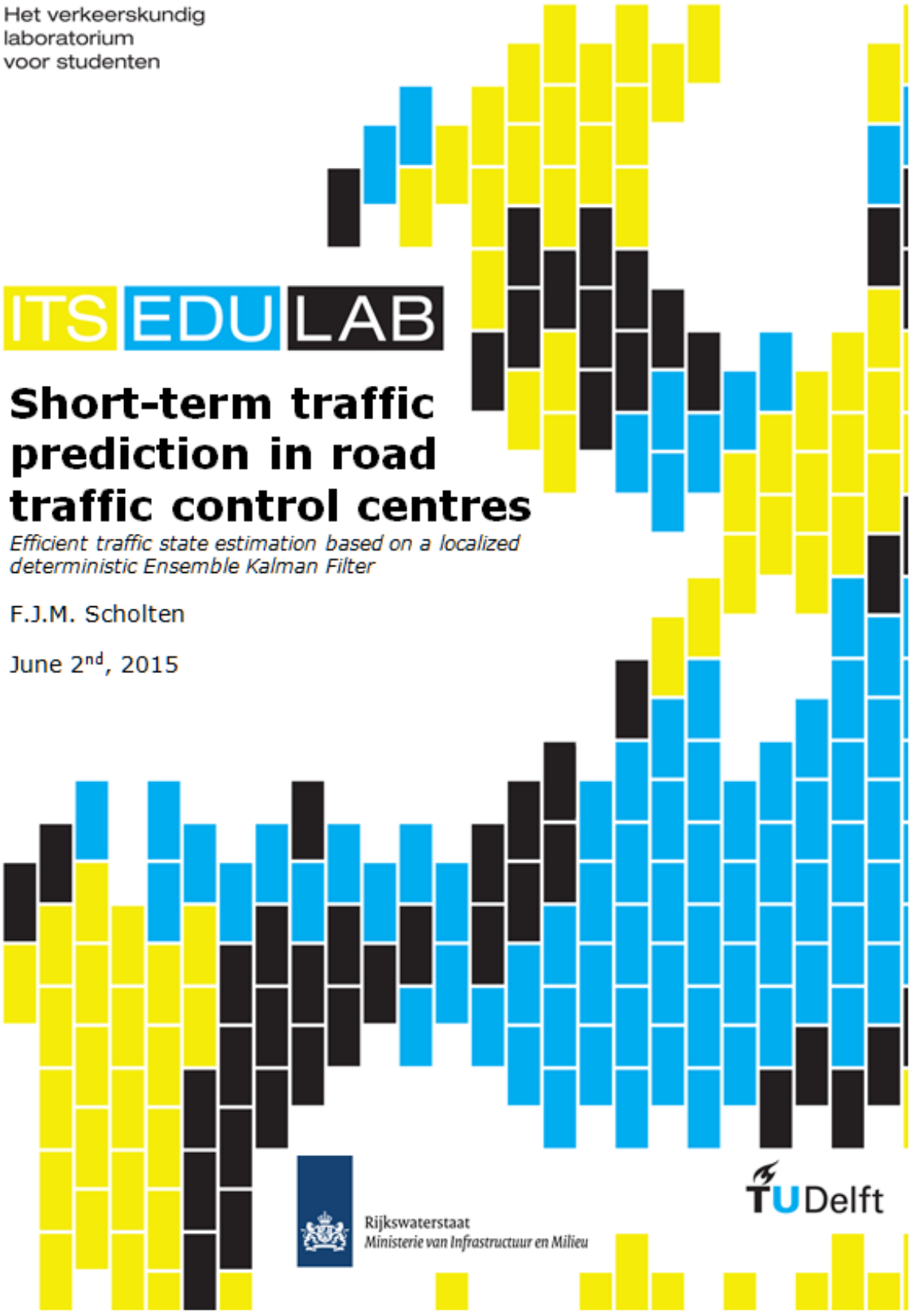

Het verkeerskundig
laboratorium
voor studenten
ITS EDU LAB
Short-term traffic prediction in road traffic control centres
Efficient traffic state estimation based on a localized deterministic Ensemble Kalman Filter
F.J.M. Scholten
June 2nd, 2015
Rijkswaterstaat
Ministerie van Infrastructuur en Milieu
TU Delft

Colophon

Author	Friso J.M. Scholten
Graduation Committee	Prof.dr.ir. Hans van Lint Dr. ir. Yufei Yuan Dr. ir. Henk Taale Dr. ir. John Baggen
Published by	ITS Edulab, Delft
Date	June 2nd, 2015
Status	Final report
Information	Henk Taale
Telephone	+31 88 798 2498

ITS Edulab is a cooperation between Rijkswaterstaat Centre for Transport and Navigation and Delft University of Technology

TRANSPORT, INFRASTRUCTURE & LOGISTICS
Delft University of Technology

Short-term traffic prediction in road traffic control centres.

Efficient traffic state estimation based on a localized deterministic Ensemble Kalman Filter.

for the degree of

MASTER OF SCIENCE

Friso J.M. Scholten

Delft, Nederland
June 2015

Contents

Summary

The current state-of-the-practice in the Dutch operational road traffic management is mostly based on experience: the traffic manager judge the traffic dynamics on a road network on basis of their previous experience and select the right control measures. As the amount of data and number of possible control measures available increase in the future, the Dutch operational traffic management needs to implement a good decision support system as the traffic managers can't handle all this (possibly conflicting) information simultaneously. This decision support system should provide a clear view of the current and near-future situation on which the right control measures can be based.

Despite some pilot studies and a lot of scientific research in traffic prediction, automated traffic estimation and predictions are not yet used in the Dutch operational traffic management.

This thesis has as objective to combine the current and near-future state of the practice with state-of-the-art knowledge in order to develop an architecture of a monitoring and prediction tool. Moreover, this thesis further examines the possible performance of such a tool by means of a prototype.

By means of a requirement analysis the most important requirements of an estimation and prediction tool are identified. From different use cases, it is derived that the estimation and prediction tool is most beneficial in non-recurrent situations. Moreover, the influence of the control measures taken by the traffic manager should be incorporated into the prediction results.

On basis of these requirements, an architecture is developed. It is chosen that the predictions should be based on macroscopic simulation models, as these models are most competent to include non-recurrent conditions and control measures. An advanced state estimation method such as (the variants of) the Kalman filter is needed to adapt the simulation model to the traffic situation at hand.

A prototype is built in order to further analyse the performance of the data assimilation method. As data assimilation method, the Ensemble Kalman Filter (EnKF) is chosen instead of the Extended Kalman Filter, that is more commonly used in the state estimation of macroscopic traffic models. The EnKF is then theoretically analysed, and three improvements to the traditional formulation are identified:

1. The main correction equation is reformulated using the Sherman-Morrison-Woodbury reformulation. This reduces the computational complexity without loss of accuracy.

2. Instead of the traditional stochastic approach, the deterministic approach of Sakov and Oke (2008) is adopted. This deterministic approach avoids the influence of coincidental sampling, at the costs of an analytical approximation of the posterior state covariance.

3. By localizing the relation between model elements that are physically distant in the real system are restricted. This localization improves the estimation accuracy as it removes fake correlations and increases the effective ensemble size.

This prototype is subjected to a number of simulation experiments in order to test the performance. Synthetic observations were used that are generated by a macroscopic traffic model. The prototype was able to estimate the traffic state reasonably well in 40 times faster than real time, when no structural differences existed between the assimilation model and the true model. Especially the localization increases the accuracy considerably: it is not feasible to get the required accuracy using a global method.

When imperfect system knowledge was assumed, the performance dropped. Further research and development of the prototype could increase the accuracy in these conditions by using additive errors instead of multiplicative errors. Further research should focus validating the model using observations from microscopic traffic models and real data. Moreover, more complex traffic models can be used.

Preface

"It is difficult to make predictions,
especially about the future."

Danish proverb

This report is the end result of my graduation thesis of the master program Transport, Infrastructure & Logistics at the Delft University of Technology. This research is performed in association with ITS Edulab, which is a cooperation between Rijkswaterstaat and the Delft University of Technology.

My gratitude goes to the members of my graduation committee. My daily supervisor Yufei provided me with useful ideas and helped me keep my focus on the task. The opportunity to join the ITS Edulab given by Henk was very useful to combine both the theoretical and the practical perspective of the traffic prediction problem. I would like to thank John for reviewing my preliminary reports. Last but not least, I would like thank Hans for his contagious enthusiasm, which inspired me to go deep into the data assimilation field.

Friso Scholten
Delft, April 2015.

Chapter 1

Introduction

In this chapter an introduction is given to the topic of this thesis: the use of short-term road traffic estimation and prediction in Dutch highway traffic control centres.

The first section gives a short introduction to the motivation of this research. The second section derives the research objectives and research questions. The third section describes the chosen methodology. The other section give the relevance and outline of this thesis.

1.1 Problem definition

As road traffic is only a self-organizing system to a certain extent, the Dutch public authorities find it as its task to influence and control the traffic operations on the Dutch roads. As the infrastructural measures such as the addition of lanes are not sufficient in combating all problems that arise with rising mobility, control of road traffic on the operational level is necessary. This observation is also shared by Rijkswaterstaat, which is the Dutch executive agency for i.a. management of road traffic on the main road network. The current focus of Rijkswaterstaat is to promote the more efficient use of the existing infrastructure.

As the responsibility of traffic managers in the Netherlands will shift from actively operating dynamic traffic management measures to monitoring the network-wide public goals and conditions, traffic managers will need more advanced decision support systems for making the right decisions on the right time (Rijkswaterstaat, 2013). The decision making process of the traffic operators is complex in several aspects, and will probably become more complex in the future.

Firstly, data issues exist: the traffic operators need to interpret lots of data coming from various sources in different forms. Examples are road-side cameras that can be controlled by the traffic operator, but also road-side systems that measure travel time on a road stretch and speed measurements at certain locations. In the future, more data becomes available, for example from individual cars that transmit their data, e.g. speed or headway, on certain time intervals. Moreover, more complex traffic control options become available. One can think about individual or vehicle class route suggestions,

network-wide coordination of ramp metering installations or dynamic speed advice. An operator has to interpret all these data and come up with the right responses.

Secondly, the impact of traffic management will become larger, as the utilization of the road infrastructure in the Netherlands increases. Traffic management will become a deciding factor as the efficiency of the self-organization of the traffic system decreases with a larger utilization of the infrastructure. The need for the Dutch operational traffic management to be effective is high.

One way for the Dutch traffic control to remain effective, is the implementation of a good decision support system. In order to select the best control option, a traffic operator needs a clear view of the current and near-future situation on the road system. Currently, the estimation and prediction of the traffic situation is mostly based on experience and tacit knowledge: traffic operators implicitly form their opinion on the traffic situation at hand and in the near future based on their experience with the job. The implementation of a tool that better monitors and predicts the traffic situation would provide large benefits to the traffic control. The operators in the traffic control centres agree with this observation: in a survey a total of 80% of the respondents that they are optimistic about the use of short-term predictors. (Mott MacDonald, 2012)

Despite some pilot studies and a lot of scientific research, automated traffic estimation and prediction is not yet used in Dutch traffic control centres. This could be caused by technical issues (e.g. incompatibility with existing systems or low performance of predictions), but also organizational issues (perceived lack of benefits or fear of change) and financial issues (high implementation costs or high operating costs).

1.2 Research questions and goals

Concluding, the application of traffic state prediction would be beneficial for the operational road traffic management, now and especially in the future. A clear gap occurs between the traffic state prediction in literature and the traffic state prediction in practice. The goal of this project is to combine the state-of-the-art theoretical basis with the current traffic management practice, in order to design (a prototype of) a traffic state predictor that clears the barriers of implementing a prediction tool.

1.2.1 Research objective

The first objective of this research is to combine the current and near-future state of the practice with state-of-the-art knowledge in order to develop an architecture of a monitoring and prediction tool that is beneficial for a Dutch regional traffic management centre. This architecture can be used as guideline for the future development of such a monitoring and prediction tool.

The second objective is to further examine the possible performance of a monitoring and prediction tool. If such a tool is capable of achieving a sufficient performance, this could

lead to further interest in such a tool from practical and research perspective.

1.2.2 Research questions

These two research objectives lead to the following two main research questions:

1. What architecture of a short-term prediction tool will be useful for the current and near-future Dutch operational traffic management practice?
   (a) What functional, performance and stakeholder requirements are imposed on a monitoring and prediction tool?
   (b) Which estimation and prediction paradigm suits these requirements best?
2. Could a monitoring and short-term prediction tool be capable of achieving a sufficient accuracy within the computation time available in a real-time setting?
   (a) Based on the chosen estimation and prediction paradigm, how should the real-time observations be optimally used in the estimation of the traffic situation?
   (b) Based on the chosen estimation and prediction paradigm, is a monitoring and short-term prediction tool capable of achieving a sufficient accuracy faster than real-time using synthetic observations?
   (c) Based on the chosen estimation and prediction paradigm, how sensitive is a monitoring and short-term prediction tool to imperfect knowledge of the real system?

1.2.3 Scope of thesis

In order to reduce the complexity and fit the project into the proposed time, choices need to be made concerning the scope of the project.

Type of road

The highways are chosen instead of arterial or urban roads due to the reduced complexity of the highways. There are many detectors present on the Dutch highways and the traffic flow is quite organized, which can lead to more accurate predictions of the traffic state. Moreover, the highways have the highest priority in the traffic management process, and would be the first application of traffic state predictors.

Complexity road network

It is chosen to look at this project on a regional network scale. This aligns with the increased focus on network management. Furthermore, this also corresponds to a scale where a prediction of 1 hour in advance makes sense: a prediction of 1 hour of a road

stretch of 2 kilometers where traffic only drives for a minute will mostly depend on the boundaries of the road stretch. However, a network scale increases the complexity as route choice will play a larger role and the number of parameters to be estimated will increase greatly.

Moreover, the choice for a regional highway network scale ensures that the influence of the need for coordination between different traffic control centres is limited.

1.3 Project approach

As there are two main research questions, this thesis is divided into two main parts where each part is focused on one main research question. The approach for answering the two main research questions are discussed in next subsections.

1.3.1 Approach design of architecture

A systems engineering approach is used to form the architecture (US Department of Defense Systems Management College, 2001). This approach consists of three major parts:

1. Requirement analysis
2. Functional analysis
3. Design synthesis

The requirement analysis is mainly based on the identification of use cases. By identifying how the monitoring and prediction tool would be used in the operational traffic management context, several issues can be deduced in each step of the use case. One could for example deduce which function the tool should provide (e.g. provide a prediction 1 hour ahead), but also in which conditions (e.g. non-recurrent traffic conditions) and the interaction with other systems (e.g. the monitoring system should give non-conflicting information with other systems).

In this research, the functional analysis and design synthesis quite overlap as the goal is a mainly functional architecture instead of a physical design. The functional analysis is used to decompose high-level functions into lower-level functions. Furthermore the (performance) requirements give direction how to design the architecture that fulfils these requirements.

1.3.2 Approach evaluating performance prediction tool

The best way to evaluate the performance of the prediction tool is to actually develop the prediction tool. Therefore, in the next paragraph the methodology how to come up with the design is selected. After that the steps of the methodology is further elaborated on.

Choice of design methodology prediction tool

In order to achieve the best possible results, it is good to think about the risks that could lead to failing the project. By investigating these risks, suitable design processes and methods can be selected.

- **Context risks:** the context of the traffic control centres is quite stable. Although there is a change in traffic management processes from old-fashioned control of traffic to supervision of traffic, the requirements of the proposed prediction tool will not be changed much. The largest risk context wise is the use of new complex technology. As this technology is hard to make and the exact benefits are unclear, it is a large risk that the new technology doesn't satisfy the requirements in terms of functions and quality. Therefore an incremental prototyping approach is suitable, as the quality of the prototype can be checked while adding functionalities.
- **Project risks:** one of the project risks is the short time frame of only 6 months. This leads to a need for sharp planning and sharp scope of the project. By using an iterative/incremental design process, the scope can be adjusted along the way when difficulties arise. Moreover, it is not possible to expect a full implementation within this short time frame. Therefore the goal is to design a prototype: this reduces the dependency on other stakeholders such as the traffic control centre.

As can be concluded from the identification of the risks, an iterative prototyping approach will be suitable for this part of the project (Pressman, 1992). Due to the short time frame of the project, only the first iteration(s) of the prototype are done. The goal of the prototype is work from simple to complex: the first version of the prototype consists of only the essential components with the simplest implementation. For example, no effort is put into visualization and security aspects.

Elaboration on incremental design steps

The incremental model consists of four parts: analysis, design, code and test. (Pressman, 1992)

Analysis In the first phase, the requirements are further analysed so the requirements are clear.

Design This part consists mostly of selecting the theoretical concepts and mathematical formulations of these theoretical concepts for making short-term traffic predictions. The goal is to use state-of-the-art techniques, such as model-based estimation (using a Kalman Filter approach) and prediction methods.

Code The code part consists of selecting the numerical algorithms and exact software formulations in order to match the design. The preference is to collaborate with a new

project of TU Delft called OpenTraffic. OpenTraffic is an open-source traffic simulator written in Java.

Test It is important to test the prototypes in their ability to recreate the results as intended by the modeller (verification) and real life (validation).

Possible verification methods are dimension analysis (checking if the equations match on the dimension of the variables used) and numerical analysis (checking if the results change much when precision of numerical algorithm is increased).

1.4 Overview thesis

In this section an overview of this thesis is given. First an outline of the contents of this thesis is given with the main conclusions and choices. After that, the scientific and practical relevance of this research is elaborated on.

1.4.1 Thesis outline

As this thesis has two main research questions, this thesis is split into two parts.

In part I a functional architecture is developed how a traffic estimation and prediction tool should be implemented and which functionalities it should have. A number of requirements are derived by analysing when traffic estimation and prediction is useful. Examples of these requirements are the applicability in non-recurrent conditions and the inclusion of the effects of control measures of traffic operators. From these requirements, a functional architecture is designed. The basis of this design is a model-based estimation and prediction approach using a Kalman Filter approach.

In part II a prototype that fits the architecture of part I is developed. Chapter 4 and 5 are literature based chapters that further analyse the main parts of the prototype. It is chosen to adopt a fairly simple system model in chapter 4. In chapter 5 the choice is made for the Ensemble Kalman Filter (EnKF) instead of the mainly used Extended Kalman Filter (EKF). The EnKF is a promising alternative to the EKF on theoretical grounds: both algorithmic as technical benefits of the EnKF exist. Further theoretical analysis of the EnKF identifies three refinements to the traditional EnKF: the Sherman-Morrison-Woodbury formula, deterministic approaches and localization. In chapter 6, the prototype is developed and verified. In chapter 7, the prototype is used for six different simulation experiments, testing the performance of the EnKF and its refinements on a large scale road network by means of a twin experiment.

After that, conclusions are drawn and recommendations are made for further research. It is shown that the localized deterministic EnKF is capable of estimating the state very fast. Although no definitive conclusions can be made about the accuracy as no validation is done using real data, the computational speed causes the localized deterministic EnKF to be a promising research direction.

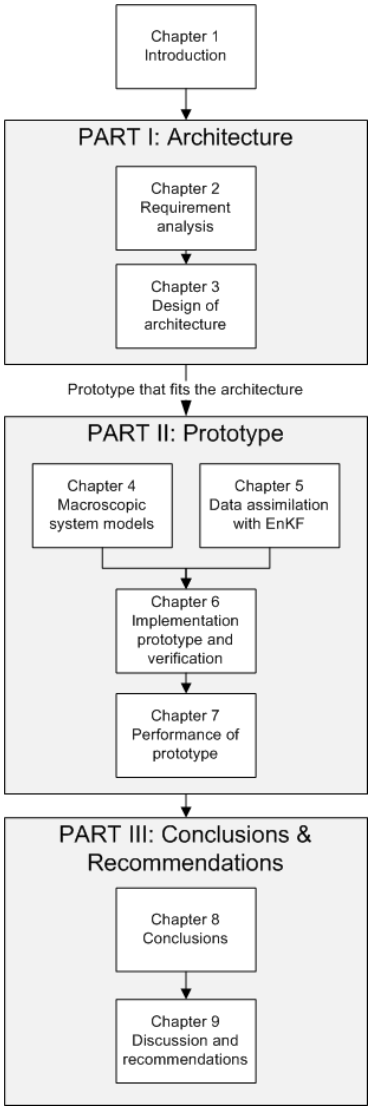


Figure 1.1: Graphical outline of the thesis.

The motivations of the different choices are further elaborated on in the corresponding chapters.

1.4.2 Scientific relevance

- *This research gives a functional architecture for traffic state estimation and prediction.* This functional architecture can be used in further research and further prototypes.
- *This research gives a quite extensive theoretical analysis of the Ensemble Kalman Filter (EnKF).* This theoretical analysis includes comparison with the EKF, reformulations for efficient computation, avoiding sampling errors and localization for increased accuracy and computational speed.

- *This research is the first application found that uses a EnKF using a first-order macroscopic traffic model with non-linear observations.* Previous research amended the traffic model to make the observations linear, restricting the generalizability to other observation types.
- *This research shows the need of localization of the EnKF approaches when applied to large scale traffic network.* In the simulation experiments of this research, the local analysis increased the accuracy tremendously in comparison to its global counterpart. The global EnKF approaches are not feasible for the use in real-time accurate state estimation.
- *This research gives empirical evidence that the DEnKF by Sakov and Oke (2008) performs better than the traditional EnKF in the use with a first-order macroscopic traffic model.* The DEnKF increases both the accuracy as the robustness to the calibrated assimilation parameters.

1.4.3 Practical relevance

- *The requirements derived in chapter 2 can be used as starting point in projects for implementation of a traffic prediction tool.* Although the requirements mostly focuses on the functional requirements that are not very case-specific, the requirements indicate the main functions and constraints that should be fulfilled.
- *The architecture developed in chapter 3 can serve as a guideline for further studies of the implementation of traffic prediction in operational traffic management.* The choices made in the architecture, such as the choice for predictions based on simulation instead of statistics, are fundamental in the design process. A head start in the development of a traffic monitoring and prediction tool will shorten the development time and increase the chance of successful implementation.
- *Efficient alternative for network-wide state estimation.* The localized (D)EnKF, that is used as data assimilation method in the prototype, is a promising solution for the traffic state estimation of large scale networks. Although the data assimilation method needs to be validated using real data, the computation time of the localized (D)EnKF makes it possible to simulate large scale networks, possibly up to nation-wide, on one computer. This way it is an efficient alternative to (an efficient implementation of) the Extended Kalman Filter.

Part I

Architecture of traffic estimation and prediction tool

Chapter 2

Requirements of traffic estimation and prediction tool

In this chapter, the requirements which a traffic estimation and prediction tool should fulfill are derived.

The first section covers the background of the Dutch operational traffic management. Here it is defined what operational traffic management is and how this operational traffic management works in practice now and in the near future. Moreover, the main stakeholders in the operational traffic management are (briefly) analysed in order to derive their main common interests and conflicts. The estimation and prediction tool should accommodate these interests and mitigate the effects of the conflicts.

The second section derives the requirements from a few use cases. In these use cases, it is described how the estimation and prediction tool would be used. From these use cases the functions and constraints of the tool are derived. Moreover, some stakeholder requirements are derived from the stakeholder analysis.

The third section compares the found requirements with the requirements of previous studies.

2.1 Background of operational traffic management

In this section the background of the Dutch operation traffic management is described. This background serves as input for the requirement analysis that follows.

2.1.1 Operational traffic management in a broader context

The increasing use of the road transportation system induces delays, environmental problems and safety concerns. Therefore, governments see a role for themselves to properly manage traffic on the roads. Before further analysing the roles the governments can take, the (road) transportation system is further analysed.

The basic layer model of Schoemaker, Koolstra, and Bovy (1999) can be used to analyse a transportation system in general, and thus the road transport system in particular. See figure 2.1 of a graphical description of this model.

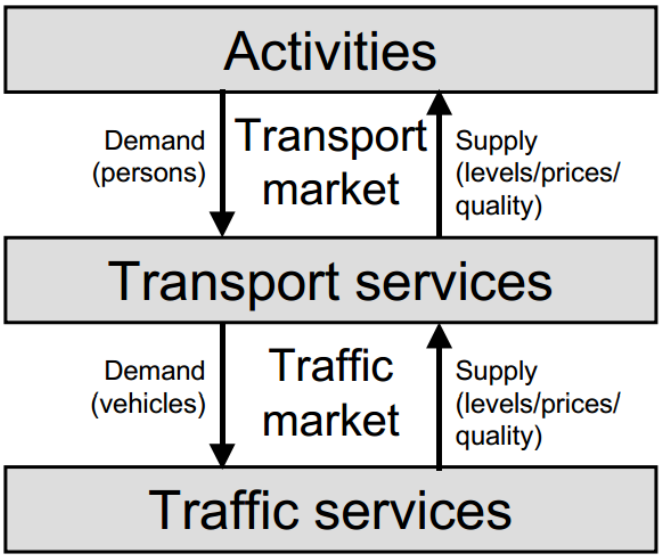


Figure 2.1: Basic layer model of Schoemaker et al. (1999), via Van Nes (2002).

The basic layer model of Schoemaker et al. (1999) consists of three layers: (economic) activities, transport services and traffic services, with the transport market and the traffic market in between.

The activities layer represents the economic activities that lead to the demand for transportation. As travellers use transportation for a reason (e.g. for commuting or transportation of goods), the activities layer indicates the reasons the users have of using transportation.

The transport services layer represents the facilities that can accommodate the need for transport generated by the (economic) activities. Examples of transport services are (privately owned) cars and lorries, but also forms of public transport. The transport services layer generates a demand pattern in space and time for the traffic market.

The traffic services layer represents the facilities that provide the infrastructure of the vehicles of the transport layer. This infrastructure accommodates the trips generated by the transport services.

The transport market and the traffic markets balance the demands and supplies. The main difference is that the transport market deals with the *distribution* of trips over mode, time and space, and the traffic market deals with the *handling* of these trips given the infrastructure.

Governments can influence the outcome of the transport and traffic markets in multiple ways. The Dutch government adopted three major strategies to properly manage road transport and traffic:

1. Building new infrastructure such as extra roads or lanes. In this way the capacity of the infrastructure is increased, which increases the supply into the traffic market. However, this strategy is very expensive and politically challenged.
2. Pricing and other demand management methods. By this strategy the government tries to manage the demand for transport. Examples of these methods are pricing the use of the infrastructure, promoting changing of working hours and promoting working at home. These methods are mostly unaccepted by public or the effectiveness unclear (to the general public)
3. Better utilization of infrastructure. This strategy refers to the promotion of more efficient use of existing infrastructure. An example of a method that focuses on better utilizing the infrastructure is the provision of better route information to the road users

Weng (2010) coupled these three strategies to the basic layer model described above, see figure 2.2.

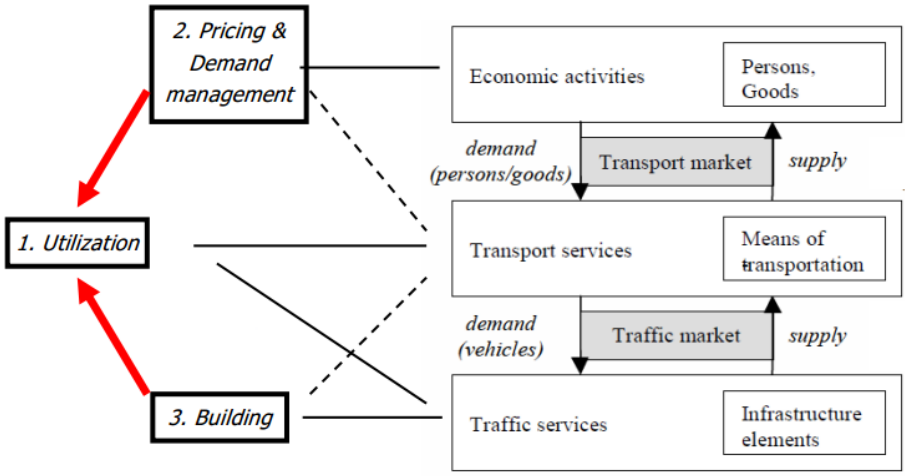


Figure 2.2: The solid lines indicate direct influence; the dashed lines indicate secondary influence. Adapted from Weng (2010).

In this research, by traffic management is meant influencing the infrastructure supply and traffic demand of the traffic market in such a way that they match best, both in time as space. The goal of traffic management is to achieve a better utilization of the infrastructure. Although at first sight the word utilization seems to only refer to the throughput or capacity of infrastructure, traffic management can also focus on making the use of infrastructure friendlier to the environment.

Traffic management is operational traffic management (as opposed to tactical or strategic traffic management) when the focus lies on the *execution* of the operational measures on a day-to-day basis. Operational traffic management is here also referred to as traffic control.

In order to translate the broad goals such as better utilization and safety to the use of operational measures on a specific network, the "Gebiedsgericht Benutten Plus" (Region-

specific Utilization Plus", GGB+) approach is mainly used in the Netherlands. This approach consists of nine steps. In general, the GGB+ approach identifies the gaps between how the network should function given the policy objectives of the different stakeholders and the actual situation on a network. From these gaps, the right solution directions and operational measures are selected. At last, the control measures are combined into a control philosophy, that indicates which operational measures should be applied.

After the suitable operational measures are identified by the GGB+ approach, control scenarios are made that are in line with the vision set by the GGB+ approach. These control scenarios are made using the "Werkboek Regelscenario's" (Instruction manual control scenarios). These control scenarios translates the the suitable operational measures of the GGB+ approach (e.g. reduce speed on route X) to the exact execution of the operational measures (e.g. set speed limit to 80 km/h on matrix signs A, B and C).

2.1.2 Taxonomy of traffic control

The traffic management (here also referred to as traffic control) can be classified on several scales. This taxonomy is given in order to provide more information of the used methods in operational traffic management and the main characteristics operational traffic management should have to be efficient.

Basic working principles

The road system is in itself a self-organizing system. However, when the utilization if the system increases, the road system fails to be as efficient as could or should be. When more demand is put on a road network, the number of vehicles able to flow out of the network decreases.

Four main principles that end the efficient self-organizing system are:

- Blocking back and grid-lock. One main issue with is that the occurring congestion takes space on the network. This occupied space can lead to problems. One example is congestion on a highway that blocks traffic routed for an off-ramp upstream of a bottleneck. This way, vehicles are delayed by a bottleneck that they will not pass. This spillback causes further acceleration of congestion.
- Capacity drop. The capacity of the road drops up to 30% when congestion sets in. More specifically: drivers tend to take more (time) distance to the vehicles in front when driving out of congestion than when the traffic is not broken down.
- Unequal spread of traffic. The unequal spread of traffic implies that more routes will contain bottlenecks. Therefore the congestion is spreading spatially as more traffic is diverted onto alternative routes. The traffic on these routes will become denser and possibly congested. This leads to more traffic that will divert, which completes the traffic-degrading circle.

- Inefficient behaviour of individual travellers. Drivers tend to make selfish choices in routes that minimize their own travel times or travel costs. However, the sum of costs associated with these individual selfish choices are higher than when the system-optimal choice behaviour occurs. This is related to game theory, i.e. the widely known prisoners' dilemma where selfish choices lead to worse overall results than cooperative choices. From an economic standpoint, the (negative) external costs (costs imposed upon a third party) are not taken into account in the route choice of an individual traveller. Specifically, the travel time losses of other drivers doesn't affect the choice behaviour of a driver.

The effects of these four principles can be mitigated by four main solution directions:

- Control spillback of queues. This solution direction implies that only traffic headed for an active bottleneck is affected by the queue.
- Improve throughput. This solution direction tries to improve the throughput by increasing the capacity where possible in order to prevent the capacity drop.
- Route guidance. This solution direction distributes the traffic more evenly in order to prevent congestion spreading. This route guidance can require giving priority to certain groups of travellers.
- Limit inflow. In this solution direction, the inflow in subnetworks are limited so that the most severely congested parts are able to recover and will not spread to other parts of the network.

The possible control measures used by the traffic management centre work according to these solution directions. Examples of possible (dynamic) operational control measures are:

- Traffic lights. One of the main urban traffic management measures are traffic lights. In recent years, the configuration (i.e. green times) of the traffic lights are made dynamic, based on traffic demand, and changeable from traffic management centres. By reconfiguring the green times, certain routes can be prioritized by the traffic manager.
- Ramp meters. By restricting the access from the on-ramps to the highway, it is prevented that congestion occurs on the highway and therefore associated problems such as the capacity drop are prevented or delayed. Moreover, drivers will possibly avoid these on-ramps on the long term, which can lead to better traffic flow. The metering rate (including obstructing the flow completely) is essential in the working of the ramp meters. If the rate of vehicles entering the highway is too high, congestion will occur and the ramp meter fails. If the rate is too low, the buffer on the on-ramp will become full too fast and the congestion will spillback onto the underlying road network or need to be "flushed". Not commonly used is mainline metering, which is equivalent to ramp metering on a main road.
- Route information panels. By informing or advising the drivers on the travel times of possible routes, one tries to redistribute the traffic more efficiently. The route information panels can be dynamic (so called DRIPs) and temporary.

- Dynamic speed limit. The speed limit can be dynamically set just upstream of congestion to improve safety as the drivers are warned and speed differences are decreased. Another way of using dynamic speed limit is to remove wide moving jams by means of the SPECIALIST algorithm.
- Peak hour lanes. The traffic operator can decide to open or close special lanes when the flow on the road is high. Two types of peak hour lanes exist in the Netherlands: a peak hour lane on the hard shoulder lane (emergency lane) and a peak hour lane on the left side of the road (which is commonly narrower than usual). As the peak hour lanes normally serves as emergency lane, the peak hour lanes are checked manually or automatically on stationary vehicles and other obstacles before and during opening. The peak hour lanes increase the throughput of the road and decrease the spillback caused by congestion.
- Dynamic lanes. On some locations, lanes can be reversed to be used by different driving directions. This is primarily useful on roads with asymmetric peak hour directions. Related are lanes that are adaptable to prioritize routes at junctions. For example, a lane can be set to accommodate extra traffic for left-turning traffic instead of right-turning traffic.

The control measures vary from informing the drivers, advising the drivers or controlling the drivers.

Geographical scale

In principle, the control measures above influence the traffic quite locally. However, it is important to consider the influence of the control measures in a regional context, as different local controls can counteract the impact of each other. Optimal regional control is more than just the sum of optimal local controls.

From the perspective of a traffic control centre, the regional scale has two major impacts on the traffic control procedures. One is that the controls should be coordinated in order to achieve the best performance. Another impact is that the regional scale implies that the traffic is controlled on different types of road, governed by different road authorities. The traffic control centre thus should account for the different priorities set by the different road authorities. The method of deriving scenarios by the "Workbook control scenarios" tries to accommodate these two issues.

Selecting control measures

Two main approaches are possible for selecting the right set of coordinated control measures by the operator.

The first, currently used, approach is the scenario based approach. This approach uses predefined scenarios: one has predetermined what are the best actions in certain cases. The traffic operator has to select the right scenario to the traffic situation at hand, manually or preselected by some decision support system.

The second approach is an optimization approach. In this approach, the deployment of the control measures is optimized using an algorithm and a traffic model. The prediction of the traffic state is an integral part of this approach, as the optimization needs a mathematical description of the traffic state or performance in the near future in order to optimize.

One main advantage of the scenario approach that this approach is quite transparent: the traffic operator sees which scenarios are assessed and can adapt the scenarios if possible. The optimization approach is more a black box approach: the algorithm gives an answer but the reasoning is quite unclear. Another advantage of the scenario approach over the optimization approach is that this approach is relatively easy to implement. On the other hand, the optimization approach can theoretically yield better results, as the fit between the traffic situation at hand and a predefined scenario isn't perfect and the scenario approach can't handle unseen conditions. The optimization approach is much better able to adapt to situations and respond optimally.

2.1.3 The future of operational traffic management

Figure 2.3 gives a concise roadmap of the future of the operational traffic management. More data sources become available that need to be interpreted by the traffic operator. Moreover, an integrated monitoring and prediction tool could be a stepping stone for further automatisation of traffic control.

The geographical scale on which the traffic is controlled could increase in the future, as the traffic management will be more effective that way.

The strategy of selecting the control measures would preferably go to a more optimization approach instead of a scenario based approach. As the number of possible control options will increase, together with more available data and higher needed effectiveness of the operational traffic management, the use of an optimization approach is needed. The proposed integrated monitoring and prediction approach is a stepping stone in the roadway to this optimization approach.

2.1.4 Stakeholders

In this subsection the main stakeholders of the operational traffic management are identified. These stakeholders are identified using Hoogendoorn, Westerman, and Hoogendoorn-Lanser (2011).

From these stakeholders, the common interests and conflict in interests are identified. By making a tool that accentuates the common interests and mitigates or resolves the conflicts between the stakeholders, the tool will have a higher chance of success.

Overview stakeholders

- *Operational traffic managers.* They are the main users of a monitoring and prediction tool. The traffic operators are responsible for the activation and deactivation of several control measures. Currently the traffic operators make their decisions mostly based on experience. They fit the traffic situation at hand to a predefined scenario and select the appropriate control measures, while implicitly considering the near-future. Moreover, traffic engineers are responsible for the generation and evaluation of the control scenarios. The current evaluation of the traffic management is quite hard as the influence of operational traffic is hard to identify and quantify due to the absence of possibility of (blind) experiments. As part of the road authorities, the operational traffic managers are most interested in efficient and safe travel on their network.

- *Private service providers.* The private service providers use the traffic information in order to use in their own products. Examples of these products are route information and navigational tools. A major factor for these private service providers is the existence of a business model: they need to cover the costs associated with their products. In their products, the private service providers are focused on the goals and wishes of their customers (e.g. the individual road users), which may contradict with the interests of the society as a whole.

- *Government.* The government has in general a broader goal than the road authorities. Not only are they interested in a efficient road network, its focus lies also on the economic position of the ITS and mobility sector, the (environmental) impacts on the neighbourhood around the roads and cost effectiveness of its activities.

- *Road users.* The individual road users are mostly interested in their own travel costs. These individual wishes could deviate from the societal goals.

- *Research groups such as universities.* These research groups have generally the most knowledge about traffic behaviour. They try to use or sell their knowledge in the operational traffic management field. Moreover, they are interested in testing their own research questions and hypotheses using real-life data and systems.

Common interests

- *Provision correct, reliable and objective information.* All stakeholders are interested in the provision of correct, reliable and objective traffic information, such as travel times and activated control measures, to road users and other interested parties. The proposed monitoring and prediction tool could provide this information to the interested parties. The private service providers can use this information to improve their products, and the road authorities try to improve the traffic flow on their network.

Conflicts

- *Cost effectiveness.* The government is more interested in the cost effectiveness of the operational traffic management than the other stakeholders. The operational traffic managers (and possibly the research institutes) are more convinced of the success of their approach. As the government constantly makes a cost-benefit analysis and the costs are generally quite clear, quantifying the benefits of operational traffic management could resolve this conflict.
- *Individual vs. collective interests.* The individual road user are most interested in the their own travel costs. As the road users are the main customer of the private service providers, the service providers try to give the best information possible to the individual road user, even if it is conflicting with the goals of society as a whole. The government can try to alleviate this mismatch by using public stimuli, such as compensation for loss of travel time or (in the future) congestion pricing. For these stimuli to succeed, the impact of this mismatch and the effects of stimuli should be quantifiable.
- *Knowledge gap.* The main centre of knowledge about traffic behaviour and operational traffic management lies with the research institutes. Although the operational traffic managers have of course much practical experience with controlling traffic, the theoretical knowledge of the underlying principles is mostly present within the research institutes. This knowledge gap leads to non-optimal adoption of new knowledge in operational traffic management. The monitoring and prediction tool should be developed and adaptable with this theoretical knowledge in mind.

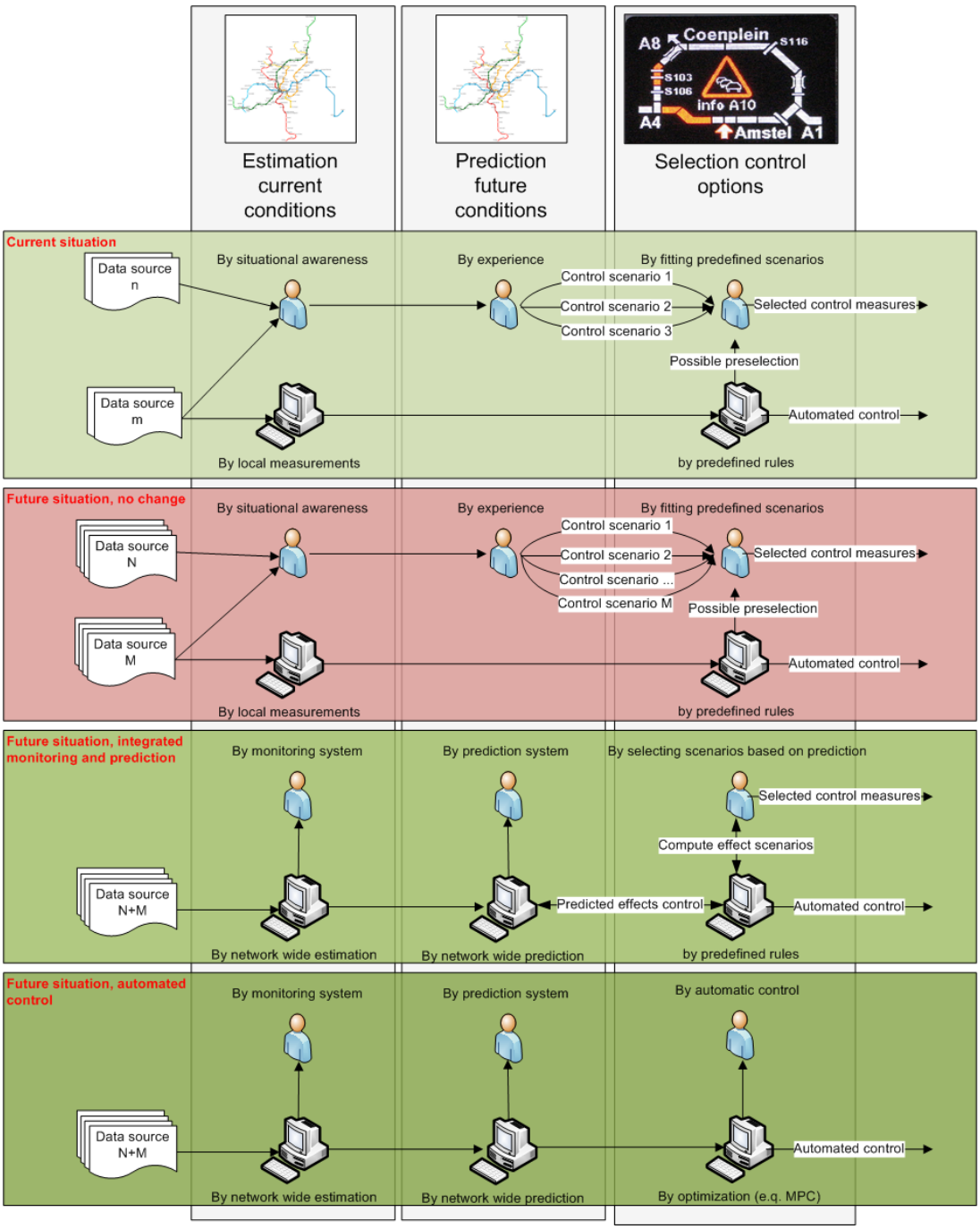


Figure 2.3: The three tasks of a traffic manager depicted against the possible traffic management scenarios. The first scenario is the current situation. The second (undesirable)scenario is the future situation when no changes in the working processes are made. The third scenario is the future situation with an integrated estimation and prediction tool. The fourth scenario is the future situation with automatic control, which is the desired situation.

2.2 Requirement analysis

In the previous section the background of the Dutch operational traffic management is described. This background gives insight in what operational traffic management is, which methods the traffic operators could use and the different stakeholders in the operational traffic management.

As concluded in the introduction and the investigation of the future trends in traffic control, it is essential that accurate and reliable traffic predictions are available to the operators in the traffic management centre. This section gives the main requirements such a tool should fulfil.

These requirements are derived from three different use cases of such a tool and the responsibilities and goals of the different stakeholders. It is chosen to derive the requirements from the use cases with at the end a comparison to the previous experience of the STEP project and the Verkeersonderneming project. The requirements are thus not directly derived from the previous projects. This choice is made in order to keep the comparison independent and valid.

2.2.1 Use cases

In this subsection, three typical use cases of a traffic estimation and prediction tool are described. The use cases are situated in the current traffic control procedures where the traffic is controlled using control scenarios. The use cases describe the way an integrated monitoring and prediction tool would be used.

Real-time monitoring during recurrent or predictable conditions

This use case is the base use case of a traffic operator. Consider a traffic operator that starts his work shift at the traffic management centre.

The first thing a traffic operator does in (the preparation of) his shift is the creation of situational awareness. The traffic operator makes an assessment of the traffic situation at hand and in the near future. The traffic operator can ask himself many questions. Is the traffic behaving normally? Are there road works that influence the traffic and the possible control measures? Are there events that require special focus in the controlled region? What are the weather predictions? The traffic operator has many options for creating this situational awareness. Not only the proposed estimation and prediction tool will provide this information, but also roadside cameras, weather information and social media such as Twitter.

The traffic operator looks not only at these information providers independently, but also compares them to each other: do they give the same information? Therefore it is important that the information is aligned in time and space, so a fair comparison can be made. This monitoring function of the traffic operator is performed throughout its shift.

The prediction function can be used to predict the impact of the (recurrent) congestion. If the impact of the applied scenarios and control measures can be modeled and used in the predictions, the traffic operator can use the prediction to determine the time of activation and deactivation of the control measures. The time horizon on which the traffic operator needs to predict is not set in stone. In general, a prediction horizon of 1 hour should be sufficient. It is important to make the operator understand how much confidence the operator can put in the prediction results.

Real-time monitoring during non-recurrent and unpredictable conditions

In addition to the previous use case, unpredictable conditions can always occur. A good example of these unpredictable conditions are accidents.

In the case that an accident occurs, the traffic operator first has to know that the accident has occurred. This can be done by for example an alert that pops up in the system, or by means of other information sources. Then the traffic operator needs to make an assessment of the impact of the accident, the time duration and the necessity of activating control measures. These control measures can be imposed by a fitting pre-existing control scenario, or can be imposed by making an ad-hoc scenario.

The prediction component can play a large role in this use case. The traffic operator can see the impact of the incident using different (ad-hoc) control scenarios. Important is that the incident length is an input to the system that needs to be supplied by the traffic operator or other information systems. With this incident length, the prediction system can perform a sensitivity analysis on this incident length: what is the impact on the traffic system when the incident influences the traffic longer?

Ex-ante or ex-post evaluation of implemented or unimplemented scenarios

As opposed to the real-time monitoring function, the system can also be used in a off-line context. Multiple off-line applications can be thought of:

- Ex-ante or ex-post evaluation of control scenarios
- Ex-ante or ex-post training of traffic operators
- Evaluation of estimation and prediction algorithms

In principle, the off-line applications can be used in two ways: firstly the estimation component itself can be replaced by a simulation component that simulates the traffic state. The second option is to simulate the detector data and the other input of the estimation component. The second option is better, as the estimation algorithms can then be evaluated. However, this requires an extra restriction on the estimation component, as the estimation procedure must provide the same results with the same input. Therefore no randomness can be involved in the algorithms, or the randomness should be controllable.

2.2.2 Functional requirements

F.1. The system should be able to monitor the traffic in real-time.

F.2. The system should be able to predict the traffic state up to 1 hour into the future

F.3. The system should be able to estimate and predict the traffic in an off-line context using simulated data.

F.4. The system should be able to correctly estimate the traffic state in both recurrent and non-recurrent situations.

F.5. The system should be able to incorporate possible control scenarios into the estimation and prediction.

F.6. The system should give reproducible results

F.7. The system should estimate and predict on all locations in the network; also locations without a detector present

F.8. The system should be adaptable to multiple different data sources

F.9. The system should be able to output multiple data variables, including number of vehicles on a road stretch, speeds and (total) delay.

2.2.3 Performance requirements

P.1. The system should provide the estimation results faster than real-time.

P.2. The system should provide a prediction under a minute.

P.3. When the system is in real-time monitoring mode, the latency in the results must be minimal or non-existent.

P.4. The system should achieve a reasonable estimation accuracy

P.5. The system should achieve a reasonable prediction accuracy in predictable conditions

2.2.4 Stakeholder requirements

These stakeholder requirements are mainly derived from the stakeholder analysis.

Derivation stakeholder requirements

The provision of correct, reliable and objective information is mainly covered by **P.4.** and **P.5.**. The information should be accessible by third parties (**S.1.**). In order to make this provision possible, the estimation and prediction tool should be user-friendly so the tool will be actively used by the traffic managers (**S.2.**)

The stakeholder analysis made clear that quantifying the cost effectiveness of operational traffic management would help the government legitimize its support of operational traffic management (**S.3.**).

In order to analyse the effects on the individual travellers as basis for further policy analysis, one should be able to evaluate the impact of the control measures on the individual travellers, which is covered under requirements **F.3.** and **F.5.**. This evaluation will most likely be based on historical data, which must be saved (**S.4.**).

In order to minimize the knowledge gap, the main architecture should be modular in the sense that main components of the architecture can be independently changed. This way new theoretical knowledge can be incorporated into the tool without disturbing the operation of the operational traffic managers by training of new systems (**S.5.**).

Overview stakeholder requirements

S.1. The information that is put into and generated by the monitoring and prediction tool should be accessible by third parties.

S.2. The user interface of the prediction tool should be user friendly.

S.3. The added value of traffic management should be (able to be) calculated automatically on a daily basis

S.4. The tool should provide a database of data of previous days so that those can be simulated and evaluated.

S.5. The architecture should be modular so the individual components should be easily modifiable.

2.3 Comparison requirements with previous studies

The found requirements are validated by comparing the requirements to previous studies on this subject, in particular the STEP project (Mott MacDonald & Fileradar, 2011) and the Rotterdam project by de Verkeersonderneming (VO)(Verkeersonderneming, 2014). In tables 2.1, 2.2, 2.3 the results of this comparison are displayed. Here the main conclusions of this comparison are presented.

This research has a stronger focus on the integrated combination of estimation/monitoring and prediction than the STEP project and the Verkeersonderneming project, which are only focused on the prediction part. In this research, the additional focus on the estimation of the traffic state is preferred due to two main reasons. The first reason is the increased trust by the users in the prediction when it is shown that the current situation is estimated correctly. The second reason is that the manual state estimation by the traffic operator will become harder in the future due to fewer cameras available and more data from individual vehicles.

The STEP and VO project impose more specific requirements on technical side and the user friendliness of the prediction tool. When implementing a prediction system, it is advised to investigate these requirements more in-depth. In this research, the focus lies more on the functional side of the traffic prediction.

The VO project focuses more on the comparison of the predicted traffic state with historical measurements, for example by means of explicit visualizing recurrent and non-recurrent congestion. This is a nice function for the traffic operator, although it is questionable if this requirement should be a "must-have" as identified in the VO project. It requires an (extensive) classification in order to distinguish between recurrent and non-recurrent congestion. This distinction can also be made by the traffic operator himself, just as in the current procedure.

The STEP project and VO project focused on the direct application of traffic prediction in operational traffic management. This research also identified use cases for a traffic prediction tool other than the use by the traffic operator. The prediction tool can also be used for ex-ante and ex-post training purposes of traffic operators using simulated data, for ex-ante and ex-post evaluation of control scenarios and the evaluation of estimation and prediction algorithms. Moreover, the prediction tool can be used for evaluating "what-if"-scenarios. This way, the added value of traffic prediction can be identified on a daily basis, which possibly helps policy makers to invest more in operational traffic management.

This research	STEP	Verkeersonderneming
F.1.	The estimation of the actual traffic state is an integral part of the STEP pilot.	The VO project doesn't consider the estimation of the current traffic state explicitly. When the predictions are compared to the actual and historical values, the crude measurements are used.
F.2.	The STEP project indicates that a prediction tool should deliver accurate results with a time horizon of at least 20 minutes. The aspiration is a longer time horizon, but reasonable results of 20 minutes into the future is the minimal requirement.	The VO project defines the prediction horizon as a multiple of 5 minutes with a minimum of 5 minutes and a maximum of 60 minutes. The predictions should be initializable manually and periodically.
F.3.	Off-line estimation is possible using the STEP approach.	An off-line aspect is not mentioned in the VO project.
F.4.	The STEP project also indicates that the prediction in non-recurrent situations is crucial.	The VO project has specific focus on the prediction in non-recurrent situations. Firstly, the prediction should be capable of incorporating capacity changes due to network changes such as incidents (requirement FE17). Secondly, it considers the difference between regular congestion and irregular congestion explicitly, and should be presented differently to the operator.
F.5.	The STEP project indicates this requirement that it should be able to simulate the control scenarios as highly desirable.	The VO project sets the incorporation of the impact of a control measure as an explicit requirement.
F.6.	The reproducibility is not mentioned in the VO project, but the end result probably satisfies this requirement.	The reproducibility is not mentioned by the VO project.
F.7.	The approach used by the STEP project considers all locations.	The Verkeersonderneming project is not very clear if it considers non-measurable locations. It seems implied in the set of requirements as the location of the queues should be accurately described.
F.8.	The used approach can use different data sources, such as loop detectors but also rain sensors.	The Verkeersonderneming project seems to focus on the data available from the NDW, which currently are the loop detector data.
F.9.	The STEP project is somewhat more specific: it indicates that both speed and travel time (and delays) should be predicted.	The VO project is very specific in the output: i.a. speed, volume, travel time, (vehicle) delay, congestion length.
-	-	The user should receive a warning when non-recurrent congestion occurs.
-	Explicit comparison of current and future traffic situation with historical average data	The current and predicted situation should be compared with historical data, e.g. weekly averages.

Table 2.1: Comparison functional requirements of this research with the STEP project and the Verkeersonderneming project.

This research	STEP	Verkeersonderneming
P.1.	No computation time used in the STEP project was found, although in order to make running the tool feasible, fast state estimation is needed.	No explicit estimation is considered.
P.2.	The STEP project has as requirement that the predictions need to be available within minutes.	The VO project has as requirement that the predictions need to be available within 30 seconds, without slowing the work process of the operator.
P.3.	The latency was a major factor in the STEP project. During the STEP pilot, the latency was reduced from 7 minutes to 3.5 minutes.	The VO project requires the used data to be as actual as possible. Therefore the prediction tool should use the MRM system ("Meetraaimanager").
P.4.	No explicit accuracy of the estimation is taken into account in the STEP project	No explicit estimation accuracy is needed.
P.5.	The STEP project is more specific in this requirements, as it indicates that both queue length and traffic speed should be accurately predicted. The required accuracy is not quantified: it should be "reasonable".	The VO project requires an accuracy varying between 70 % and 80% depending on the prediction horizon and the predictability of the congestion.

Table 2.2: Comparison performance requirements of this research with the STEP project and the Verkeersonderneming project.

This research	STEP	Verkeersonderneming
S.1.	No explicit requirement, although due to the web based technology it is possible for third parties to access the data.	The VO project requires that the prediction tool should be independently used by both the regional traffic operator and the highway traffic operator. By using a webbased application, also third parties outside Rijkswaterstaat should be able to view the results of the traffic prediction tool.
S.2.	The STEP project further specifies the user friendliness by means of live trials. It indicates that color-coded links on a map view were an essential view. Moreover, a dual display of both the current situation and an animation of the predictions was very successful	The VO project requires user friendly visualization of the predictions. Its requirements on this topic are mostly based on the STEP project.
S.3.	No mention of comparing with "what if" scenarios is made.	No mention of comparing with "what if" scenarios is made.
S.4.	The STEP project used an extensive database of historical traffic data.	The VO project requires a database consisting of the traffic data of 1 year.
S.5.	The tool used in the STEP project is quite modular.	No explicit mention is made about the modifiability of individual components, as new insights can be incorporated into new versions of the prototype.

Table 2.3: Comparison stakeholder requirements of this research with the STEP project and the Verkeersonderneming project.

2.4 Conclusions

In this chapter, the Dutch operational traffic management was analysed, culminating in a set of requirements in subsections 2.2.2, 2.2.3 and 2.2.4. These requirements were derived by describing use cases how the traffic prediction tool was used. Moreover, the interests of different stakeholders are identified in order to make the implementation of a traffic prediction tool successful.

The most important requirements are that the prediction tool should deliver accurate results in a real-time setting. These results should be accurate in non-recurrent conditions, as the prediction tool is most relevant to the traffic operator in these conditions. Moreover, the prediction tool should be able to incorporate control measures taken by the traffic operator.

The requirements derived in this thesis are compared to the requirements in the STEP project and the Verkeersonderneming project. The different projects agree on the most important requirements, but differ slightly on some details. This research focuses more on the integrated combination of estimation and prediction instead of only prediction,

as it is perceived that the state estimation is a quite hard problem and accurate state estimation leads to more trust in the prediction results. Another difference is that this research identified more use cases, such as training and evaluation purposes, of a traffic prediction tool. These additional use cases can lead to more enthusiasm by the decision makers as the added value of the traffic prediction tool would be higher and can be made explicit.

Chapter 3

Design of architecture

As in the previous chapter multiple requirements for an integrated monitoring and prediction tool are derived, in this chapter an (functional) architecture is produced that satisfies these requirements.

Here the definition of Clements et al. (2002) for a architecture is used:

> **Architecture:** the set of structures needed to reason about the system, which comprises elements, relations among them, and properties of both.

As a basis for the architecture, the control cycle is selected. The elements of the control cycle are further specified in the following sections using the state-of-the-art. Firstly it is argued that a model based prediction approach is most suitable. Then the macroscopic model type is selected as traffic flow model paradigm. In the third section the estimation approach is selected. In that section, the Kalman filter approach (belonging to the recursive Bayesian method class) is selected as main component.

Section 4 gives a more detailed overview given the choices in the first sections. In section 5, the match of the architecture with the requirements of the previous chapter is investigated.

3.1 Prediction approach

A lot of research is done in short-term predicting the traffic state. Van Hinsbergen, Van Lint, and Sanders (2007) give a taxonomy of short term traffic prediction models used in literature. They consider short-term traffic prediction as solving an input-output problem, where the input (with some parameters) is transformed by a model into an output.They divide the different methods into naive, parametric (also known as model-based) and non-parametric (also known as data-driven) methods. Naive methods are methods that use no (or a very simple) model and no parameters deduced from data. Parametric methods use a model with a predetermined model structure, and used some parameters from (real-time) data. The non-parametric methods also derive the model structure from the data. Other taxonomies divide the possible approaches in similar

classes: the non-parametric class is also known as the empirically based (Arem, Kirby, Vlist, & Whittaker, 1997) or machine learning (Nikovski, Nishiuma, Goto, & Kumazawa, 2005) class; the parametric class is also called the traffic process theory based (Arem et al., 1997) or the dynamic traffic assignment (Nikovski et al., 2005) class.

3.1.1 Naive prediction

Naive models are models that do not use any model assumption. It can be interpreted as the use of only the data at hand and exact physical relationships (e.g. distance = speed $\times$ time) (Van Hinsbergen et al., 2007). Also the (direct) use of measured variables as proxy for unmeasured variables falls within the naive approaches. Examples of naive methods are: the use of historical averages of a certain traffic variable as predictor for the future state; and the calculation of (instantaneous) travel times assuming the prevailing traffic conditions to be constant. Advantages of the these naive methods is that the calculations are non-existent or very fast and the reasoning is easy to understand by practitioners. As the accuracy of these naive methods is mostly low for short term traffic prediction, they are not considered a good alternative for this application. Note that the second example of the use of instantaneous travel times as proxy for predicted travel times is still the most widely used method for most en-route travel times in many countries including the Netherlands, although more intelligent approaches outperform this methods. (Van Lint & Van Hinsbergen, 2012)

3.1.2 Non-parametric or data driven prediction

The non-parametric approach derive the model parameters and model structure from the collected data. Examples of these non-parametric approaches are the use of regression methods, (advanced) time series or neural networks. The derived input-output relation can be characterized as a (merely) statistical relationship: no theoretical knowledge or assumptions from traffic flow theory is used. The non-parametric approach is thus a “black box”: the exact relationship between input and output is unknown or has no value for further interpretation. The advantage of a non-parametric approach is that the dynamic and non-linear traffic processes can be (quite accurately) modeled. The non-parametric approach has some disadvantages: firstly, a lot of data is needed to “train” the model. Secondly, handling of unseen scenarios that are not present in the calibration data is hard as the model is only derived from data. Thirdly, there is limited experience in application of non-parametric approaches on a network scale. Most applications focused on predicting traffic on a single location or route. Fourthly, the non-parametric approaches are very inflexible in terms of location. As the non-parametric models are trained using local data, the whole training procedure must be reapplied when considering other networks or changes in the network.

3.1.3 Parametric or model based prediction

The parametric approach has a certain model structure as starting point, and derives the parameters from the historic or current data. The model structure is chosen using theoretical analyses of traffic flow. The chosen models can range from simple analytical formulas to estimate travel time (such as the BPR function) to full-fledged microscopic traffic flow models that model every individual vehicle in the network. As this model-based approach is essentially a "white-box" approach, and therefore very suitable for traffic predictions in unseen situations or controlled cases. A large disadvantage is the vast number of parameters and variables that need to be set (correctly). This tuning requires a lot of (real-time) data that is possibly not available. Another problem is the fit of the chosen model: the assumptions that are made in the model can be very strong and unrealistic, or the model can't provide behaviour that is seen in the real world.

3.1.4 Conclusion: model-based prediction

A parametric, also called model-based, approach seems to be a wise choice, especially with requirements **F.4.**, **F.5.**, **F.7.**, **F.8.** and **F.9.** in mind. Van Lint and Van Hinsbergen (2012) suggest a hybrid approach of using parametric and non-parametric models to combine internal model variables with real-life data. This hybrid approach, where a simulation model is used as base and non-parametric models are used to estimate parameters in this simulation model, is chosen in this thesis.

3.2 Types of traffic flow models

As in the previous section a model based prediction approach was selected, the type of traffic model used needs to be selected. In this section, two main classes of traffic models are elaborated on: the macroscopic traffic models and the microscopic traffic models. Other, less commonly used, types of traffic models exist, such as the Network Transmission Model that describes traffic flow on a (sub)network level, and mesoscopic models that are in a sense a hybrid version of the macroscopic and microscopic models.

3.2.1 Microscopic traffic models

In microscopic traffic models the base elements are the individual vehicle-driver combinations (or "vehicles" in short). For these individual vehicles, the interaction with the other vehicles is described: e.g. the braking of a vehicle depending on the surrounding traffic. The basic variables used are the speed, the (time) headway and the space headway of the vehicle.

As microscopic traffic models describe individual vehicles, the resulting traffic patterns as congestion regions and traffic waves are *emergent* behaviour. Microscopic models are therefore very useful to model how single vehicles affect the traffic, e.g. for investigation

of the influence of adaptive cruise control to the traffic flow or the impact of different driving styles to traffic capacity.

Many microscopic models are stochastic, which means that random behaviour is included. In order to achieve representative results, the results of a microscopic model are usually average over multiple model runs.

3.2.2 Macroscopic traffic models

Macroscopic models describe the traffic flow analogously to liquids or gases. In contrast to the microscopic traffic models, macroscopic traffic models don't describe individual vehicles. Instead, the macroscopic models use (locally) aggregated variables for a road section.

These basic variables can be the density k which describes how close in space vehicles are apart; the flow q which describes how close in time vehicles are apart; and the average speed u of the vehicles on a road section.

Macroscopic traffic models thus describe the collective behaviour instead of the individual behaviour of vehicles. Therefore, the macroscopic traffic models are suitable when one is interested in this collective behaviour.

3.2.3 Choice of traffic model type

The macroscopic traffic model is preferred, due to the following reasons:

1. The traffic manager is mostly interested in collective behaviour, so a model that describes the collective behaviour suits the goal best.
2. Although a microscopic model may describe the traffic flow better in ideal situations, microscopic models have lots of parameters that need to be correctly calibrated. In a real-time context, this is very hard or impossible.
3. The computation time of a macroscopic traffic model is lower, as fewer degrees of freedom are used.

3.3 Estimation approach

The basis of estimation is to estimate the traffic state, which is input of the prediction component, using the traffic data from the sensors. The estimation deals with multiple main issues:

- The traffic data is measured in a different variable than the traffic state. An example is that the speed is measured instead of the density.
- Traffic data can be fused from different data sources.

- The traffic data is observed on different spatiotemporal locations as the traffic state. The traffic state could be on given on a 50 meter, 2 second interval, as opposed to measurements that are given in a 500 meter, 1 minute interval.

- The traffic data has errors. Traffic measurements seldom produce no errors: e.g. loop detectors miss vehicles or count vehicles double. Moreover, traffic data can have a structural error, also called a bias. An example of this bias is the averaging of the speed of vehicles over time, as representation of the speed of the vehicles over space.

Three estimation techniques are discussed here: naive estimation, the adaptive smoothing method and the recursive Bayesian methods. Other techniques exist, e.g. nudging. Nudging is a technique that, in the same way as the recursive Bayesian methods, combines a traffic model with detector data. However, this technique is somewhat less flexible than the recursive Bayesian methods, and therefore is omitted here. (Schreiter, 2013)

3.3.1 Naive estimation

The naive class consists of traffic state estimation methods that do not use (or very simple) model assumptions. The main advantage of these techniques are that they are very fast to compute.

One method is simple interpolation/extrapolation in space. This means that e.g. the speed on an unobserved location is estimated by copying the data of the nearest detector, or interpolation the traffic data of the detectors around that location. These interpolation methods are mostly not very effective, as the dynamics of traffic are not taken into account.

Other more elaborate "naive" methods can fuse data from several data sources in order to form data with less error; this is so-called data-data consistency. For example speed data can be fused with (realized) travel time data in order to remove the bias from the speed data. (Ou, Van Lint, & Hoogendoorn, 2008)

3.3.2 Adaptive smoothing method

The Adaptive Smoothing Method (Treiber & Helbing, 2002), extended and generalized by Van Lint and Hoogendoorn (2010), is in essence an approach that interpolates over space and time. It takes traffic dynamics explicitly into account. The basis of this method is that some characteristics, such as speed and flow, travel with a certain speed along the freeway. This propagation speed is however dependent on the traffic state: in free flow the characteristics travel downstream, but in congestion the characteristics travel upstream. The ASM interpolates the data along this propagation speed.

3.3.3 Recursive Bayesian methods

As opposed to the adaptive smoothing method, the recursive Bayesian methods combine the data of the sensors with predicted data from a traffic model. These methods are also called sequential estimation algorithms, as these methods consist of iteratively updating the estimate when new sensor data becomes available.

The name of these methods refers to Bayes' theorem:

$$P(H|E) = \frac{P(E|H)}{P(E)} \cdot P(H) \tag{3.1}$$

Loosely speaking, the (posterior) probability of a hypothesis given some evidence is determined by the likeliness of the (prior) hypothesis and the likelihood of observing the evidence with the hypothesis. In this context, the hypothesis H (traffic state) is estimated using the evidence (observations from detectors).

Commonly used recursive Bayesian estimation techniques are the Kalman filter (and its numerous extensions and adaptations) and the particle filter. The particle filter uses a (large) number of simulation runs to estimate the distribution of the posterior traffic state. However, this technique is too computationally expensive when considering high-dimensional systems such as a large traffic network. The Kalman filter assumes that the model error is Gaussian distributed, which makes the Kalman filter far more efficient if that assumption is valid. Therefore the particle filter is omitted in further considerations.

3.3.4 Comparison Adaptive Smoothing Method and Kalman Filter

In this subsection the ASM and the KF are compared. The comparison is made on both theoretical basis and the set requirements. The theoretical comparison is based on the comparison by Schreiter (2013). In table 3.1 the results are summarized.

Category	Criterion	ASM	Kalman Filter
Characteristics	Paradigm	Data driven	Hybrid of data driven and model based
	Traffic flow theoretical basis	Shockwave theory	Generic, any process and observation model
	Data estimated	Density, speed, flow	All parameters needed, including e.g. FD parameters
	Network topology	Motorway	All networks; urban and motorway
	Calibration complexity	Easy	Hard
	Prediction	Independent	Integrated
Requirements related	Inclusion control scenarios	Complex	Natural
	All locations	Yes, interpolated	Yes, physical representation by model
	Different data sources	All data sources	All, excluding time aggregated measurements

Table 3.1: Comparison Adaptive smoothing algorithm and Kalman Filter approach

Comparison of theoretical characteristics

The Kalman Filter approach is a more generic approach than the ASM. The ASM is a purely data driven approach, whereas the KF uses a data driven approach in combination

with a traffic flow model. The ASM has shockwave theory as theoretical basis, whereas the Kalman Filter approach can work with any process and observation model. (Schreiter, 2013)

The Kalman Filter approach integrates the estimation and prediction more than the ASM. This is visible in two ways: firstly the prediction is a integral part of the prediction-correction scheme used in the KF approach. Secondly, the KF approach ensures that all state elements and parameters needed for predicting using the traffic flow model are estimated, as the same model is used for estimation and prediction. If the ASM is used, additional parameters have to be (independently) estimated in order to complete the necessary set of parameters for the prediction model.

The ASM approach is only suitable for motorway traffic. The Kalman Filter approach can also use a (macroscopic) urban traffic model. (Schreiter, 2013)

The main problem of the Kalman Filter approach is that for correct estimation, a lot of parameters need to be calibrated correctly. The ASM is less complex due to the fewer degrees of freedom. (Schreiter, 2013)

Comparison on requirements

When the focus lies on the requirements derived in the previous chapter, the Kalman Filter scores slightly better. The inclusion of the influence of control in the state estimation is more natural in the Kalman Filter approach than in the ASM. Take for example a case where (possibly class-specific) rerouting is implemented by the traffic operator. In the Kalman Filter approach, the estimation of the (class-specific) density is quite natural, whereas in the ASM this information is hard to add.

The Kalman Filter is also slightly better in the estimation of the traffic state in unobserved locations than the ASM, as the estimated state between detectors is estimated by the model instead of just smoothed value of the detected values. This is graphically explained in figure 3.1. In this figure, a situation is described where congestion has formed between two measurements. The ASM smoothes the measured values, which omits the congestion inbetween. The traffic flow model used by the Kalman Filter predicts the congestion as result of the lane drop.

The ASM is capable of incorporating time aggregated measurements, such as realized travel times. It isn't computationally feasible for the Kalman Filter approaches to incorporate these travel times, as the (augmented) state vector becomes too large. (Van Lint & Hoogendoorn, 2010)

3.3.5 Conclusion: Kalman Filter approach

The Kalman Filter approach is far more generic than the ASM approach. This means that complex system models and different data sources can be used in without completely changing the framework. Moreover, the Kalman Filter approach integrates the estimation

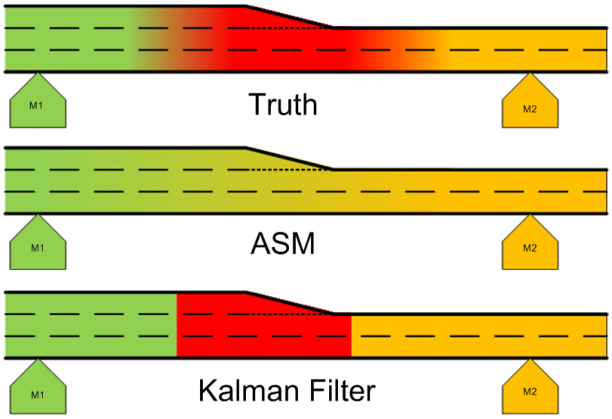


Figure 3.1: Possible traffic pattern near a lane drop and associated congestion between two loop detectors. The ASM approach interpolates the measurements, which omits the congestion. The traffic flow model in the Kalman Filter can correctly predict the occurrence of congestion even though the measurements don't indicate the congestion.

and prediction part. This is advantageous, as it ensures that the estimation and prediction part are aligned.

The main disadvantage of using the Kalman Filter approach is the calibration procedure and associated validity. It is very hard to correctly calibrate the parameters in the Kalman Filter approach due to the large number of parameters. The desired accuracy and validity of an approach is only achieved when the parameters are set right. As the ASM needs far less parameters, this calibration is easier.

Although the list of approaches above seem to indicate that these approaches are disjoint, the approaches can be combined. One can for example use the ASM to fuse speed and travel time information together as input for a Kalman Filter.

3.4 Overview functional architecture

In figure 3.2 the derived architecture is shown.

The most important properties of the components is already described in the previous sections. Here, the structure of some components is further described.

3.4.1 Model component

In this architecture, it is chosen to use the same model for the estimation and the short-term predictions. This makes it easier to initialize the prediction model, as the state and structure of the estimation model and prediction model are compatible.

A generic model step of instant t_1 to t_2 of cell j consists of three phases (Van Wageningen-Kessels, 2013):

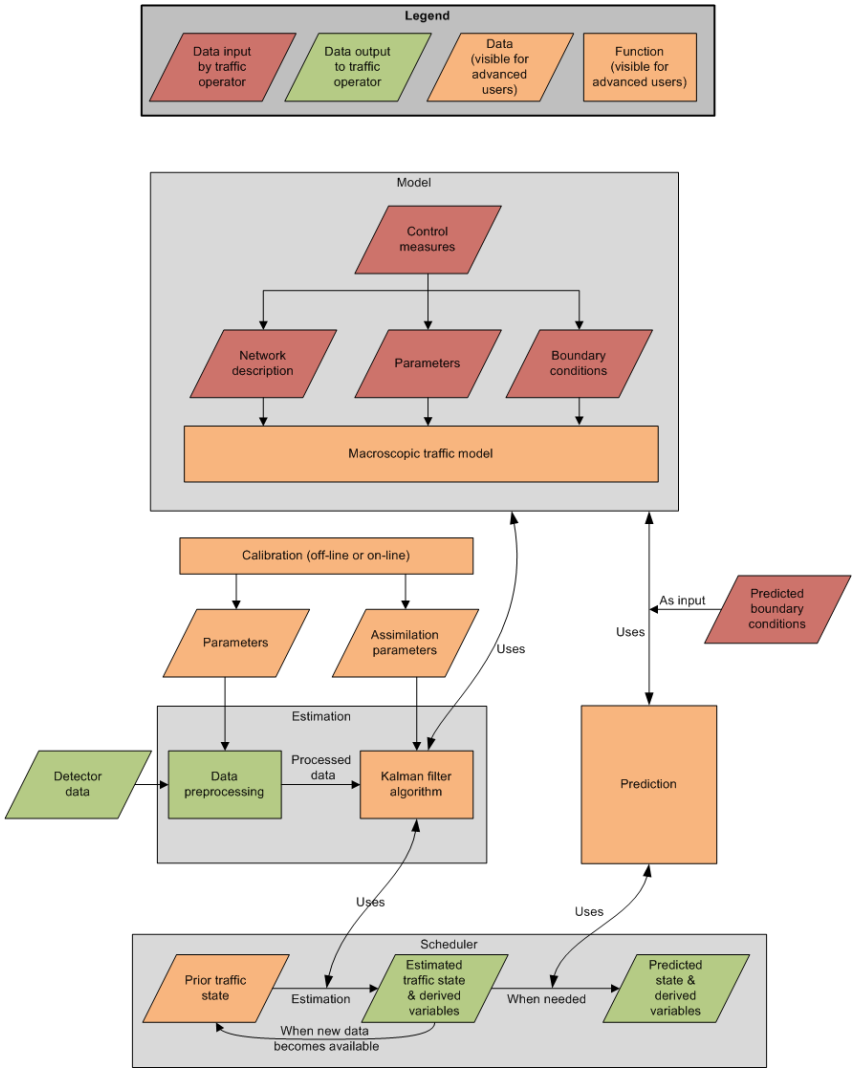


Figure 3.2: Architecture

1. Preparation: the preparation phase consists of calculating values of some variables of cell j, given the current state at instant t_1, that are necessary for determining the state at time t_2. In the context of a macroscopic traffic flow model, this step consists for example of determining the flow through a node. The preparation phase can use (and mostly will use) information about other cells than cell j.

2. Actual time step: this phase determines per simulated object (e.g. cell or node) the state of that simulated object at instant t_2. In this context this step consists of e.g. determining the density (for a LWR-model), class-specific density (for Fastlane) or density and velocity (for a Payne-model). The update of the state of cell j only uses variables of cell j.

3. New values other variables: this phase determines the values for the other variables at instant t_2. In this context this could consist of determining the speeds or pce-values. The variables that are updated are variables at cell j that only depend on the state of cell j.

The structure of the model above guarantees that the dynamics of the model don't rely on the order of execution. This becomes especially clear in the second phase. If in the second phase the update of the state of cell j was dependent on the state of e.g. cell i, the state of cell j would differ if the state of cell i was already updated or not.

3.4.2 Estimation component

The proposed estimation approach is a two-step approach. First the detector data (from conventional loop detectors, but also data from individual cars, travel time cameras or even weather information) is preprocessed. After this procession, the processed data is fed into the Kalman filter algorithm.

Several algorithms can serve (together) as data preprocessing. These processing algorithms vary from simple (removing erroneous data, such as unfeasible data values) to quite advanced (reducing observation error by the Adaptive Smoothing Method), as described in section 3.3. These algorithms can not only change the values of the detector data, but also fuse data from several detectors together and add (estimated) data on other spatiotemporal locations. The data processing algorithms need some kind of parameters, e.g. threshold values or assumed traffic behavioural values such as characteristic wave speed.

The processed data is used as observation in the Kalman Filter approach. The Kalman Filter component uses a traffic model, a prior estimation of the traffic state and the processed observation data, in order to make a (posterior) estimated traffic state. Depending on which Kalman Filter method is chosen, one or more instances of the traffic model is needed.

3.4.3 Prediction component

The prediction component is quite small. It uses the estimated traffic state and a macroscopic traffic model to predict the future traffic state. Essential for the performance is the inclusion of the correct future control measures and future parameters such as inflow and turn fractions.

3.4.4 Scheduler component

The main scheduler is the main function in the architecture. The scheduler handles the main time loop the simulation must follow. It keeps track of a queue of processes that need to be called at a certain point in time. The main process is the calling the estimation component when new observation data becomes available. Moreover, it keeps track of when a short-term prediction is needed.

The scheduler can compare the timestamps of the detector data and the real-life time to calculate the latency. The latency should be minimal. One solution to minimize this latency is to use the prediction component to predict the "real-time" data based on

the data that is a few minutes old. This solution is used in the estimation system in Düsseldorf. (Gentile & Meschini, 2011)

3.5 Verification architecture with requirements

In this section the architecture is verified. This means that it is checked if the architecture corresponds with the requirements. See table 3.2 for an overview of these requirements.

Requirement	Satisfied	Motivation
F.1.	✓	The monitoring aspect is performed by the estimation component.
F.2.	✓	The prediction aspect is performed by the prediction component.
F.3.	✓	The off-line aspect can be used by feeding artificial data into the estimation component
F.4.	✓	The model-based nature of the architecture makes it possible to be used in both recurrent and non-recurrent situations.
F.5.	✓	The different control scenarios can be used as input to the descriptions of the infrastructure and (prior) knowledge on driving behaviour.
F.6.	?	The reproducibility of the results depends on the implementation: the macroscopic traffic model and the data assimilation components should be deterministic, or incorporate stochastic methods with the use of a random number generator with an easily modifiable seed.
F.7.	✓	In principle the macroscopic traffic model can estimate and predict on all locations in the network. However, due to the discretization of the model in implementation the traffic state at some location can be approximated by the traffic state at a location nearby.
F.8.	✓	Observations from data sources can be explicitly used in the main Kalman filter algorithm. Another option is to fuse multiple data to one better data set in the data processing component. An example of this is the PISCIT algorithm that fuses individual travel time measurements and aggregated speed data from loop detectors in order to remove the bias from the speed data. (Ou et al., 2008)
F.9.	✓	Assuming that the required output can be formed by the traffic state
P.1.	✓	Practical experience with (localized) Kalman filter estimation has shown that some Kalman filter algorithms can deliver fast enough results on a comparable time scale.
P.2.	✓	Once the current traffic state is estimated by the estimation component, the prediction component shouldn't take too much time.
P.3.	?	The amount of latency depends on the used data source. Normally the data from the NDW has a few minutes delay before arriving in the traffic management centre. Two main options to combat the latency can be used. Firstly data can be used from the Meetraaimanager which data has less delay. A second option is to predict the "real-time" data based on the data of a few minutes old. This last option is used in Düsseldorf. (Gentile & Meschini, 2011)
P.4.	?	The question remains if the right accuracy can be found. It is mostly based on the geographical scale and the trade-off between accuracy and computation time. Practical experience seem to suggest that it should be possible. Further experimenting could provide this answer.
P.5.	?	It depends on the predictability of the boundary conditions. Further experimentation should provide light on this requirement.
S.1.	?	Depends on the implementation and security settings.
S.2.	✓	Depends on the implementation, but the architecture doesn't prevent user friendliness
S.3.	✓	One could use the system to predict the performance both with and without control by the traffic operator. The accuracy of the computation of this added value is quite arbitrary, as one has to assume the response of the traveller in both situations.
S.4.	✓	Logging is part of the architecture.
S.5.	✓	The modular approach of this architecture makes it possible to modify components of the architecture: e.g. use a different Kalman Filter algorithm.

Table 3.2: Verification architecture with requirements

The architecture complies reasonably well with the set requirements. The compliance with some requirements, such as the amount of latency and the prediction accuracy, rely on the exact implementation of the architecture or needs more research.

3.6 Conclusions and further steps

In this chapter an architecture was derived from the requirements in the previous chapters. On basis of theoretical arguments is is chosen to adopt a simulation model based approach

instead of a pure statistical approach. This approach is more suitable for the estimation and prediction in non-recurrent conditions and the effects of control scenarios can be evaluated. The simulation model should apply a macroscopic paradigm instead of a microscopic paradigm to limit the number of parameters and the computation time.

In further implementation of this architecture several choices need to be made. Some of these choices that need further research are:

1. *Choice of macroscopic traffic model.* Several macroscopic traffic models exist with different characteristics. In chapter 4 a short overview of different choices in the selection of a macroscopic traffic model is given.

2. *Choice of Kalman Filter algorithm.* Several variants of the Kalman Filter algorithm exist that are suitable in different situations. Further research is needed to select the right algorithm for the problem at hand. In chapter 5 the Ensemble Kalman Filter is further analysed.

3. *Data preprocessing techniques.* It needs to be further researched which combination of data preprocessing techniques are suitable in conjunction with the Kalman filter approach.

4. *Calibration procedures.* The parameters needed for the data preprocessing and the Kalman Filter need to be set with appropriate values. These calibration can be done off-line (independently of the traffic situation at hand) or on-line. Moreover, a choice need to be made which parameters need to be estimated by the calibration procedure. One has for example multiple options for the setting of the capacity of a certain link: *a)* set the capacity at a fixed value; *b)* on-line estimation of the capacity by a independent calibration method; or *c)* on-line estimation of the capacity as part of the state of the Kalman filter.

In the next part of this thesis, a prototype is made on basis of the architecture of this chapter. The main focus will lie on the selection of a right Kalman Filter approach.

Part II

Development of prototype

In part I an architecture of a traffic estimation and prediction tool was designed. In this part of the thesis a prototype is designed that fits the proposed architecture. This prototype could be seen as a first stepping stone for further more elaborate prototypes.

The outline of part II is as follows. In chapter 4 the used framework for the macroscopic system model, which represents the traffic flow, is chosen. Chapter 5 further investigates the data assimilation part of the framework. The choices made in these chapters are mostly based on theory and previous research.

Chapter 6 describes the implementation of the prototype. In the implementation part many choices are made how the selected system model and data assimilation methods are implemented. The last section of the implementation chapter the prototype is verified. Verification means checking if the prototype is correctly implemented as designed. If no errors occur, the confidence in the behaviour of the prototype is increased.

In chapter 7 a total of six simulation experiments are performed using the prototype. These experiments indicate the possible performance of the prototype: does the prototype deliver satisfying results that would justify further effort into maturing the prototype into a more elaborate traffic estimation and prediction tool?

Chapter 4

Macroscopic system models

In chapter 3 it was derived that the architecture should consist of a model-based prediction and estimation approach. In this chapter the model used in the prototype is derived. The system model consists of a process model, which describes the propagation of traffic, and an observation model, which describes the used observations.

4.1 Choice of process model

This section is divided into three subsections. The first subsection covers the choice of a coordinate system. The second subsection treats the choice of the order of the traffic model and the inclusion of different vehicle classes. The third subsection chooses the used fundamental diagram.

4.1.1 Coordinate system

Traffic flow can be analysed in three dimensions: space, time and vehicle number (Makigami, Newell, & Rothery, 1971; Laval & Leclercq, 2013). Therefore, three two-dimensional coordinate systems can be formed: the space-vehicle-number coordinates, space-time coordinates, the vehicle-number-time coordinates (Yuan, 2013). The second and third coordinate systems are also called the Eulerian and Lagrangian coordinate system respectively.

The Eulerian coordinate system is the most prevailing coordinate system in this context. A (discretised) Eulerian coordinate system is easy to understand: viewing the network as a collection of cells of fixed length, which is updated every (fixed) time step, is quite natural. The use of the Lagrangian coordinate system in traffic engineering is a more recent development. As opposed as the Eulerian coordinate system, (platoons of) vehicles are observed as they move through space. The Eulerian method can be visualized by observing the traffic standing besides the road, and the Lagrangian coordinate can be visualized by observing the traffic from within a vehicle that moves through the network.

Yuan (2013) proposes the use of the Lagrangian method, as it found that the Lagrangian method is a more accurate and efficient simulation of freeway traffic, and the Lagrangian method is more a suitable approach for the application of data assimilation methods such as the Extended Kalman Filter due to availability of a better numerical solution. However, the implementation of the Lagrangian method and an associated assimilation scheme is quite hard. Platoons need to be generated and deleted from the model as they reach the begin and end of the described network. Therefore, the state vector in the data assimilation scheme changes over time.

Therefore, the Eulerian method is chosen in this research.

4.1.2 Traffic classes and order of traffic model

Traffic models can also be classified into their use of traffic classes and the order of the traffic model. Schreiter (2013) gives an overview of different types of macroscopic traffic models:

1. macroscopic model LWR
2. mixed-class generalizations of LWR
3. multi-class macroscopic models with fixed pce
4. multi-class macroscopic models with dynamic pce

First order model: the LWR model

Macroscopic models are models that represent average traffic behaviour as a fluid. The main idea behind macroscopic models is the conservation of vehicles: vehicles can't be generated or destroyed in the model. The first macroscopic model was the LWR model independently proposed by both Lighthill and Whitham (1955) and Richards (1956). This model can be mathematically described as

$$\frac{\partial k}{\partial t} + \frac{\partial q}{\partial x} = 0, \tag{4.1}$$

which represents the (Eulerian) conservation of vehicles equation, with $q = Q(k)$ (referred as the fundamental diagram) and $q = kv$. k represents the vehicle density, q the average flow and v the average speed. Equivalently, one could also define $v = V(k)$ and $v = \frac{q}{k}$.

For numerically solving this partial differential equation (PDE), the conservation of vehicles equation is discretized in the space and time dimension. The space dimension then consists of cells of length Δx_i with $i = 1 \dots n$ with n the total number of cells. The time dimension is discretized into time segments of Δt. The discretized PDE can be written as:

$$k_{\tau+1}^i = k_\tau^i + \frac{\Delta t}{\Delta x_i} \left(q_\tau^{i-1 \to i} - q_\tau^{i \to i+1} \right), \tag{4.2}$$

for all cells i. This discretized equation has a clear structure: the density at the new time instant is the old density, plus the traffic that flows into the cell, minus the traffic that flows out.

Different numerical schemes are developed to calculate the flux $q^{i\to i+1}$, which is assumed to be maximized. The most widely used numerical scheme for this application is the Godunov scheme, also called the minimum supply-demand scheme (Lebacque, 1996). It defines at time instant τ:

$$q^{i\to i+1} = \min(D^i, S^{i+1}) \tag{4.3}$$

$$D^i = \begin{cases} Q(k^i) & k^i < k_C^i \\ C^i & \text{otherwise} \end{cases} \tag{4.4}$$

$$S^i = \begin{cases} C^i & k^i < k_C^i \\ Q(k^i) & \text{otherwise} \end{cases} \tag{4.5}$$

k_C^i denoted the critical density of cell i, and C^i the capacity (the flow attained at the critical density) of cell i. The subscripts τ were omitted for notational purposes.

The numerical method is only stable when the Courant-Friedrichs-Lewy's (CFL) conditions is satisfied, which reads

$$\Delta x_i \geq \max_k \left| \frac{\partial Q(k)}{\partial k} \right| \Delta t, \text{for all } i. \tag{4.6}$$

Most (realistic) density-flow relations $Q(k)$ attain their steepest ascent or descent at zero density, which reduces the steepest slope to the maximum velocity: $\max_k \left|\frac{\partial Q(k)}{\partial k}\right| = v_{max}$. The CFL-condition thus can be interpreted as that within a time step a vehicle can only cross at most one cell boundary. (Van Wageningen-Kessels, 2013)

At merges and diverges one has to choose how to distribute the traffic flow over the incoming and outgoing links. Tampère, Corthout, Cattrysse, and Immers (2011) proposes a generic class of first order node models for nodes with an arbitrary number of incoming and outgoing cells. However, the associated algorithm is quite complex and will cost a lot of computation time. Therefore the used network is restricted to nodes of $1 \to 1$ cells (one-to-one), $2 \to 1$ cells (merges, e.g. on-ramps) and $1 \to 2$ cells (diverges, e.g. off-ramps). The mentioned merge and diverge models are the degenerate versions of the generic model of Tampère et al. (2011).

For the diverge model one considers the turn fraction γ^i to be the fraction of vehicles arriving at the node going to link i. Assumed is that vehicles flow first-in-first-out over the node regardless of their destination. If delays occur as one of the outgoing links is congested, both outgoing links are restricted in such a way that the turn fraction is maintained. The total flow over the node is then given as $q = \min\left(D, \frac{S^1}{\gamma^1}, \frac{S^2}{\gamma^2}\right)$. The flows over the node to the different links are $q^1 = \gamma^1 q$ and $q^2 = \gamma^2 q$.

In the merge model, one has to define distribution factors to indicate the priority in merging. Here, based on Tampère et al. (2011), these distribution factors are based on

capacity: $d_i = \frac{C_i}{\sum_i C_i}$. If one defines $S_i^* = d_i S_i$, then one can define $s_i = S_i^* + \max(S_{i'}^* - D_{i'})$, where i' is the other link than link i, and $q_i = \min(D_i, s_i)$. What this means is that the supply of the outgoing link is filled maximally, where the available supply to each incoming link is based on the capacity of the incoming link. When one incoming link doesn't use all the supply available, this supply can be used by the other incoming link.

Mixed-class generalizations of LWR model

As the LWR model is very basic, some unrealistic behaviour is produced by the LWR model. For example, the LWR model assumes that the new equilibrium velocity is directly attained after a changed traffic state. This implies that vehicles in the LWR model are capable of infinite acceleration. Another example is that the transition from free flow regime to the congestion regime always occurs at the same density. (Van Wageningen-Kessels, 2013)

These problems can be mitigated by using variants of the LWR model, e.g. by introducing bounded-acceleration or using a stochastic model. Another approach is the use of a second order model. Noteworthy is the Payne model (Payne, 1971), which combines the conservation of vehicle equation with an equation for the velocity dynamics:

$$\frac{\partial v}{\partial t} + v\frac{\partial v}{\partial x} = \frac{v^*(k) - v}{t_{\text{relax}}} - \frac{c^2}{k}\frac{\partial k}{\partial x}, \tag{4.7}$$

in which $v^*(k)$ is the equilibrium velocity from the fundamental relation. The Payne received some (theoretical) criticism by Daganzo (1995) as the negative velocities could be reached in certain conditions, due to the lack of *anisotropy*. Anisotropy means that the characteristics of the traffic state can travel faster than the vehicles themselves. After that, some models are proposed that repairs this issue. Well-known is the model by Aw and Rascle (2000), which replace the Payne (Payne, 1971) velocity equation by

$$\frac{\partial}{\partial t}\left(v + p(k)\right) + v\frac{\partial}{\partial x}\left(v + p(k)\right) = 0. \tag{4.8}$$

In this equation, $p(k)$ represents a 'pressure' term. In further research, this model is adapted, extended and generalized many times.

Mixed-class vs. multi-class

The previous mixed-class models only accounted for one type of vehicles. However, different vehicle classes such as trucks, vans and person cars have different characteristics. A truck is for example longer and slower than a normal car. Multi-class macroscopic models explicitly include different vehicle classes.

The conservation of vehicles equation of the LWR model is generalized to each vehicle class u:

$$\frac{\partial k_u}{\partial t} + \frac{\partial q_u}{\partial x} = 0. \tag{4.9}$$

In this equation, the class-specific flow $q_u = Q_u(k_{tot})$ is determined by the class-specific fundamental diagram based on the total density of all vehicle classes. This represents that the vehicles of the different vehicle classes are conserved and flow according to their own (fundamental) relation. However, all vehicle classes share the same space on the network, so the flows and velocities are based on the combined total density k_{tot}. The total density k_tot can be modeled as the weighted sum $k_{tot} = \sum_u \pi_u k_u$, where the weights π_u are the passenger car equivalent (pce) values.

Most researches assume the pce values π_u having state-independent values. However, this assumption is quite crude. In free-flow conditions, the difference in physical lengths trucks and cars don't have a large influence on the average space the vehicles occupy. However, when traffic is standing still, a large truck can occupy up to 3 times the space of a passenger car. The pce values are thus state dependent. The Fastlane model explicitly uses the state-dependent pce values.

Explained phenomena

	macroscopic LWR model	mixed-class generalizations of LWR model	macroscopic multi-class model fixed pce	macroscopic multi-class model dynamic pce
congestion	✓	✓	✓	✓
spillback	✓	✓	✓	✓
congestion dissolution after increase of demand	✓	✓	✓	✓
congestion dissolution after increase of supply	✓	✓	✓	✓
capacity drop	–	✓	○	○
emergence of stop-and-go waves	–	✓	○	○
propagation of stop-and-go waves	✓	✓	✓	✓
multi-class	–	–	✓	✓
dynamic pce value	–	–	–	✓
deterministic model	✓	✓	✓	✓

Figure 4.1: Overview of phenomena explained by the different macroscopic models. From (Schreiter, 2013).

Conclusions

Although the number of explained phenomena by the single class LWR-model is low, it is chosen as model type in this prototype. This is due to a number of reasons:

1. Number of parameters. The number of dynamic parameters involved in the single class LWR-model is low: the whole system model is defined by the density k when the fundamental diagram parameters are considered constant. As the number of cells is high given the geographical scale, the number of parameters per cell are very important for the computational speed of the data assimilation algorithm.
2. Computational speed of model steps. As this model is the simplest, the computation will be the fastest. This is advantageous in this application as more time is available for data assimilation.
3. Ease of implementation. Given the limited time for the implementation this prototype, traffic models that are easy to implement are preferred.
4. Prototype approach. It makes sense to begin with the simplest approach for the first prototype, as the verification of a simpler model is easier. In following versions of the prototype more complex traffic models can be used.

4.1.3 Fundamental diagram

A critical part of the LWR model is the relation between the density and flow $Q(k)$. This relation is called the fundamental diagram. The fundamental diagram has two applications (Van Wageningen-Kessels, 2013). One application is the use of the relation as part of the traffic flow model, as described in this subsection. The other application is the use of the fundamental relation for relating observations with the traffic state.

One of the first fundamental relations was proposed by Greenshields, Channing, Miller, et al. (1935), based on seven observations. His fundamental relation consists of a linear relationship between the density and speed, which implies a parabolic relationship between density and flow. Later empirical findings indicate that at larger densities the relationship between the density and flow is approximately linear.

Daganzo (1994) proposes a bilinear relationship between the density and flow. It is also mentioned as a triangular fundamental diagram due to the shape in the density-flow plane. The linear relation between the density and flow in the free flow branch implies a constant velocity of the vehicles when more vehicles are present on the road. The interaction between vehicles in free flow is thus ignored.

The fundamental diagram of Smulders (1990) is a hybrid version of the fundamental relations of Greenshields et al. (1935) and Daganzo (1994). It assumes a parabolic relationship between the density and flow in the free-flow branch and a linear relationship in the congested branch.

The fundamental diagram of Smulders (1990) is chosen as fundamental relation in this prototype. See Van Wageningen-Kessels (2013); Li (2008); Del Castillo (2012) for more

detailed overviews and requirements of shapes of the fundamental relation.

4.2 Choice of observation model

Just as the traffic propagation model, traffic observations can be divided into two main categories: Eulerian observations and Lagrangian observations.

Eulerian observations are observed at a fixed point in space, as opposed to Lagrangian observations that are observed moving along the vehicle stream. Examples of Eulerian data are the common double loop detectors, but also cameras and radar detectors. GPS or other tracking devices transmitting data of individual vehicles are examples of Lagrangian detectors.

Incorporating Lagrangian data generally improves the estimations in comparison to using only Eulerian data. In the use with a Eulerian process model, the Lagrangian source data is usually transformed to Eulerian formulated observations that easily fit into the Eulerian process model. Here the assumption is made that the Lagrangian sensing data at that spatiotemporal location represent the conditions in a fixed cell around that location. As an example: a car transmits its position and velocity at a certain point in time, which is a Lagrangian observations. The assimilation model uses this information as a 'virtual' detector at that (fixed) location that detects the velocity. This virtual detector is thus a Eulerian observation. Essentially, the change in position of the moving detector is ignored.

Only Eulerian double loop detectors are chosen, as this is the main available measurement instrument in the current situations. These loop detectors give the velocity and the flow at a fixed location, averaged over 1 minute.

Chapter 5

Data assimilation using the Ensemble Kalman Filter (EnKF)

In this chapter the data assimilation part of the prototype is further specified. Data assimilation techniques are techniques that combine a priori knowledge, mostly consisting of a parametric model that describes a system mathematically, with real-life observations. As identified in chapter 3, the recursive Bayesian methods known as the Kalman Filters is chosen as main algorithm of the data assimilation or state estimation component.

In this chapter first a short introduction to the Kalman Filter is given. Then two different approaches, the Extended Kalman Filter (EKF) and the Ensemble Kalman Filter (EnKF), and their differences and previous applications are further examined. There it is concluded that the EnKF is a promising method. Therefore, the EnKF is further analysed for its use in the traffic state estimation and prediction prototype.

5.1 Introduction to Kalman Filter approaches

Kalman (1960) set the basis of modern filtering theory in his seminal paper, which provided a sequential algorithm to compute an optimal estimator of the state for linear discrete dynamical systems, under additive white Gaussian process and observation noise. His estimation technique estimates the state of a system optimally given observations of the system and knowledge of the system. Kalman filtering is widely applied in many different fields of engineering.

A dynamical system can be written in a state-space form:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k) + \boldsymbol{w}_k \tag{5.1}$$

$$\boldsymbol{y}_{k+1} = \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{5.2}$$

If the functions $\boldsymbol{f}$ and $\boldsymbol{h}$ are linear, the state-space equations can be rewritten by using

transformation matrices as replacements for the linear functions:

$$\boldsymbol{x}_{k+1} = F_k \boldsymbol{x}_k + \boldsymbol{w}_k \tag{5.3}$$
$$\boldsymbol{y}_{k+1} = H_k \boldsymbol{x}_k + \boldsymbol{v}_k \tag{5.4}$$

The state-space form consists of a process equation (5.1) and an observation equation (5.2) (or equations (5.3) and (5.4) in the linear case). In the equations above, the vector $\boldsymbol{x}_t$ represents the state at time t. The matrix F_k is the state transition model, which defines the function that maps the previous state $\boldsymbol{x}_k$ at time k to the new (predicted) state at time $k+1$. The vector $\boldsymbol{y}_t$ represents the measurements at time t. The matrix H_k defines the observation model, which maps the state into (predicted) measurements. $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are the noise terms. It is assumed that these noise terms are white Gaussian error terms, which means that have a zero-mean normal distribution (with covariance matrices Q_k and R_k respectively).

As the Kalman filter is a sequential Bayesian filter, it iteratively updates the state when new observation data becomes available. The Kalman filter has a clear predictor/corrector scheme. In the prediction component, the time is updated (e.g. from time $t-1$ to t) and a rough estimate of the state $\hat{\boldsymbol{x}}_t^-$ is given, called the prior state. This prior state is then corrected using observations to form the more accurate posterior state $\hat{\boldsymbol{x}}_t$. In figure 5.1 this scheme is visualized.

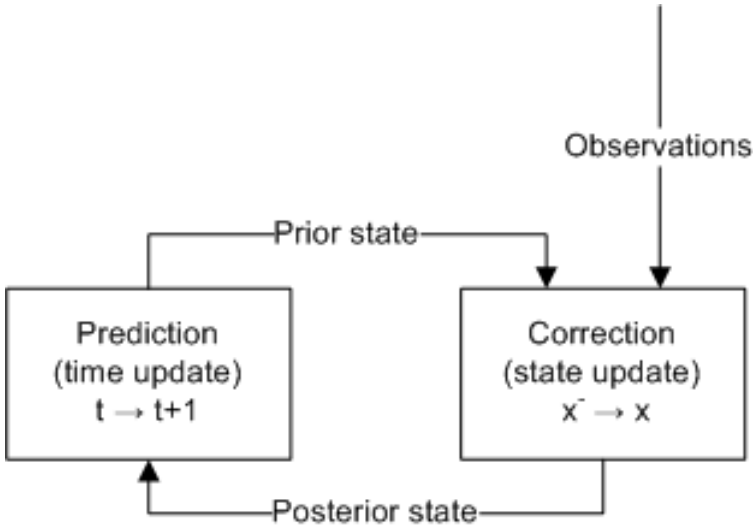


Figure 5.1: Predictor-corrector scheme of the Kalman filter approaches

Now the Kalman filter algorithm is described, using the formulation of Van Lint and Djukic (2012). Step 1a and 1b correspond to the prediction step, step 2a and 2b correspond to the correction step.

Algorithm 1: the Kalman Filter

Consider a linear space state model as in equations (5.3) and (5.4), where $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are independent, zero-mean, Gaussian noise processes with covariance matrices Q_k and R_k. Initialization:

$$\hat{\boldsymbol{x}}_0 = \mathrm{E}\left[\boldsymbol{x}_0\right] \text{ and } P_0 = \mathrm{E}\left[\boldsymbol{x}_0 - \mathrm{E}\left[\boldsymbol{x}_0\right]^\intercal\right]$$

for $k = 1, 2, \ldots$ **do**

Step 1a: predict mean and variance of state variables (forecast step)

$$\hat{\boldsymbol{x}}_k^- = F_k \hat{\boldsymbol{x}}_{k-1}$$

$$P_k^- = F_k P_{k-1} F_k^\intercal + Q_{k-1}$$

Step 1b: predict output variables

$$\hat{\boldsymbol{y}}_k^- = H_k \hat{\boldsymbol{x}}_k^-$$

Step 2a: compute Kalman gain

$$K_k = \frac{P_k^- H_k^\intercal}{H_k P_k^- H_k^\intercal + R_{k-1}}$$

Step 2b: update mean and covariance

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + K_k(\boldsymbol{d} - \hat{\boldsymbol{y}}_k^-)$$

$$P_k = (1 - K_k H_k) P_k^-$$

end

In step 1, the prior state $\hat{\boldsymbol{x}}_k^-$ at time k is predicted using the posterior state $\hat{\boldsymbol{x}}_{k-1}$ at time $k-1$ and our knowledge of the system (i.e. the model). In step 2, this prior state is corrected using the observations to form the posterior state. The Kalman gain has an intuitive structure:

$$\frac{\textbf{uncertainty process model} \cdot \textbf{sensitivity state to observations}}{\textbf{uncertainty observation model + uncertainty observations}}$$

The main assumptions of the Kalman filter are:

- the process and observation models are linear
- the noise terms are independent unbiased Gaussian with known covariance matrices

As the process and observation models in a macroscopic traffic model are not linear, the basic Kalman filter is not useful (Blandin, Couque, Bayen, & Work, 2012). Therefore multiple (non-optimal!) extensions to the basic Kalman filter are developed and applied in traffic estimation applications (Blandin et al., 2012). Here two of these extensions are investigated: firstly the Extended Kalman Filter (EKF), as the EKF is the method used most for traffic applications, and secondly the Ensemble Kalman Filter (EnKF)

which could provide better results than the EKF. Other not discussed extensions are the unscented Kalman filter (UKF) (Julier & Uhlmann, 1997; Mihaylova, Boel, & Hegyi, 2006) and the mixture Kalman filter (MKF) (Chen & Liu, 2000; X. Sun, Muñoz, & Horowitz, 2004)

5.1.1 The Extended Kalman Filter (EKF)

The basic Kalman Filter assumes that the process and observation models are linear. However, traffic flow models are not linear (Blandin et al., 2012). The EKF tries to solve this by linearising the process and observation models.

Algorithm 2: the Extended Kalman Filter

Consider a (non-linear) state space model as in equations (5.1) and (5.2), where $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are independent, zero-mean, Gaussian noise processes with covariance matrices Q_k and R_k.
Initialization:

$$\hat{\boldsymbol{x}}_0 = \mathrm{E}\left[\boldsymbol{x}_0\right] \text{ and } P_0 = \mathrm{E}\left[\boldsymbol{x}_0 - \mathrm{E}\left[\boldsymbol{x}_0\right]^\intercal\right]$$

for $k = 1, 2, \ldots$ **do**

Step 1a: predict mean and variance of state variables (forecast step)

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1})$$

$$P_k^- = F_k P_{k-1} F_k^\intercal + Q_{k-1}$$

with $F_k = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k-1}}$

Step 1b: predict output variables

$$\hat{\boldsymbol{y}}_k^- = \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-)$$

Step 2a: compute Kalman gain

$$K_k = \frac{P_k^- H_k^\intercal}{H_k P_k^- H_k^\intercal + R_{k-1}}$$

with $H_k = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k^-}$

Step 2b: update mean and covariance

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + K_k(\boldsymbol{d} - \hat{\boldsymbol{y}}_k^-) \tag{5.5}$$

$$P_k = (1 - K_k H_k) P_k^-$$

end

Notice that the due to the linearisation the EKF is not an optimal filtering technique as the basic KF per se. The accuracy of the approximation is strongly dependent on how non-linear the process and observation models are. In the context of the macroscopic

traffic models, this depends on the state: e.g. the fundamental diagram is quite linear in a free-flowing or congested state, however around capacity the linear approximation is not so good.

Aside from the state vector $\boldsymbol{x}$, which can represent the traffic system uniquely (e.g. density at every cell), several parameters that govern the dynamics of the traffic model (e.g. fundamental diagram) need to be estimated. These parameters are not explicitly taken into account in the equations above. Three main options for estimating the parameters are used in literature :

1. The parameters as input to the data assimilation. This means that the parameters are outside the scope of the EKF. This implies that the parameters are assumed constant or are updated in another way.

2. The parameters are part of the state. This is the approach preferred by Wang and Papageorgiou (2005). This way, the parameters are jointly updated with the rest of the state, i.e. cell densities using the same data. This is the most general approach. However, this means that the state vector becomes large and would seriously hamper the computation time. Moreover, the question remains if the filter can estimate the state accurately due to the high number of degrees of freedom.

3. Another option is a hybrid version of the previous two options: dual filtering. In this way, one couples two different EKF algorithms: one that updates only the state elements, and the second that updates only the parameters. For the parameter updating, one could use a different spatiotemporal scale. This agrees with the physical phenomenon: the average driving behaviour, which is caught within the fundamental diagram, varies on a broader spatiotemporal scale than the vehicles themselves.

5.1.2 The Ensemble Kalman Filter (EnKF)

Another approach to the extension of the KF to non-linear situations is to use a Monte Carlo sampling approach: the Ensemble Kalman Filter (EnKF). The idea is to represent the state distribution by using a collection of state vectors (called an ensemble), instead of using the mean state vector ($\boldsymbol{x}_k$) and the state covariance matrix (P_k).

The main equations are quite similar for the EnKF as for the other Kalman filters (Mandel, 2009). Let n be the number of state variables, m be the number of measurements and N the number of ensemble members. Instead of the vectors $\boldsymbol{x}$ (size n) and $\boldsymbol{d}$ (size m), we now use the matrices X ($n \times N$) and O ($m \times N$). The columns of X now form a sample of N members of the prior distribution. By using the main update equation ($X_a = X + K(O - HX)$), the columns of X_a form a random sample of the posterior distribution. The EnKF is completed by substituting the state covariance P in the Kalman Gain matrix $K = (PH^\intercal)/(HPH^\intercal + R)$ by the sample covariance $C = AA^\intercal/(N-1)$ with $A = X - E(X)$ (the anomalies matrix).

Algorithm 3: the (traditional) Ensemble Kalman Filter

Consider a (non-linear) state space model as in equations (5.1) and (5.2), where $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are independent, zero-mean, Gaussian noise processes with covariance matrices Q_k and R_k. Consider the observation matrix H_k given for each k.
Initialization:

$$X_0 = \mathrm{E}\left[X_0\right]$$

for $k = 1, 2, \ldots$ **do**

Step 1a: predict mean and variance of state variables (forecast step)

$$X_k^- = \boldsymbol{f}(X_{k-1}) = \left[\boldsymbol{f}(\boldsymbol{x}_{k-1}^1), \boldsymbol{f}(\boldsymbol{x}_{k-1}^2), \ldots, \boldsymbol{f}(\boldsymbol{x}_{k-1}^N)\right]$$

$$P_k^- = \frac{AA^\intercal}{N-1}$$

$\boldsymbol{x}_t^i$ denotes the i-th ensemble member at time t, and $A = X_k^- - \mathrm{E}\left[X_k^-\right]$ is the anomalies matrix
Step 1b: predict output variables

$$Y_k^- = H_k X_k^-$$

Step 1c: perturb observations

$$O = [\boldsymbol{d} + \boldsymbol{\epsilon}_1, \boldsymbol{d} + \boldsymbol{\epsilon}_2, \ldots, \boldsymbol{d} + \boldsymbol{\epsilon}_N]$$

Step 2a: compute Kalman gain

$$K_k = \frac{P_k^- H_k{}^\intercal}{H_k P_k^- H_k{}^\intercal + R_{k-1}}$$

Step 2b: update mean and covariance

$$X_k = X_k^- + K_k(O - Y_k^-) \tag{5.6}$$

end

5.1.3 Applications of the EKF and EnKF in macroscopic traffic simulations

Application of the EKF

Quite a lot of research is done using the EKF for state estimation in macroscopic traffic models, both on theoretical basis (Wang & Papageorgiou, 2005) and large scale case studies. The EKF is used for all kinds of traffic models, varying from second order mixed-class models to multi-class models. Wang and Papageorgiou (2005) provide a general approach of using the EKF in macroscopic traffic state estimation.

Special attention is given to the localized Extended Kalman Filter (L-EKF) by Van Hins-

bergen, Schreiter, Zuurbier, Van Lint, and Van Zuylen (2012). The basis of the L-EKF is that the covariance between elements of the state $\boldsymbol{x}$ that are physically distant is, and should be, close to zero. As consequence, a measurement at a specific location has a negligible influence on the state at a location far away from this location. The L-EKF exploits this feature by using a measurement of a detector to only correct the states of cells in the vicinity of that detector. In the L-EKF algorithm many EKF-analyses are done sequentially, using only one measurement and only the cells in the neighbourhood of the measurement at a time.

The main advantage of using the L-EKF is the computational speed when considering a large network. This can be seen by investigating the EKF equations: the $\left(H_k P_k^- H_k^\intercal + R_{k-1}\right)^{-1}$ is now a 1×1 matrix instead of a $m \times m$ matrix. As the inverse operation is notoriously slow (and inaccurate), the transformation of to the inverse operation of a scalar value saves a lot of time. Moreover, all matrix multiplications are faster due to the reduced size of the matrices.

Application of the EnKF

Only a few articles have been found that an EnKF approach for data assimilation in macroscopic traffic models.

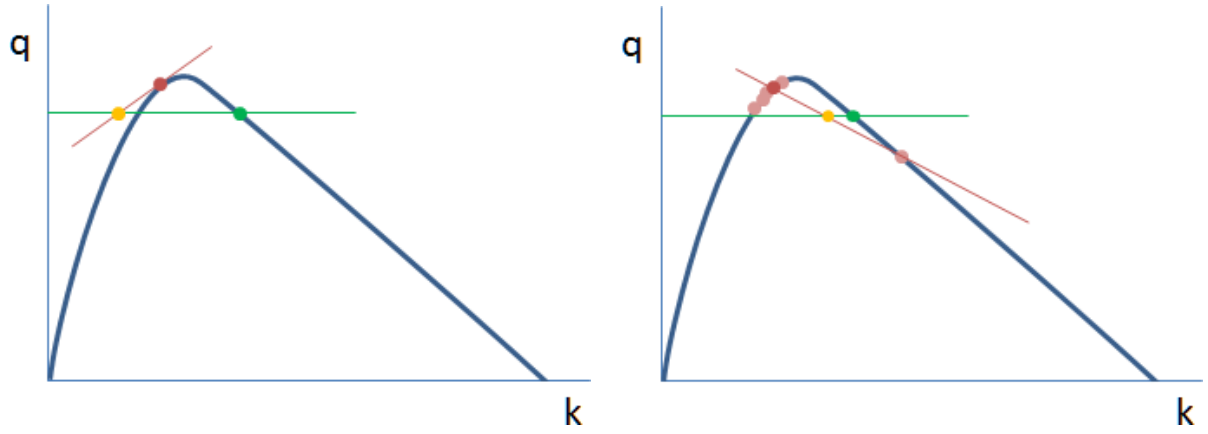
Work et al. (2008) used instead of the chosen LWR model, where the densities of the cells are used as state, the adapted LWR-v model where the velocities of the cells are used as state. In this way, they avoid using a non-linear observation function H_k as the observations are directly linked to state elements. Coric, Djuric, and Vucetic (2012) employed the EnKF is the same manner as the Work et al. (2008): also the speed version of the LWR model is used. In a follow-up article, Work, Blandin, Tossavainen, Piccoli, and Bayen (2010) used the EnKF with a localization approach to make sure that the framework is suitable for large scale application.

In this thesis, it is opted to use a more general approach where different process models (e.g. standard LWR, possibly multi-class) or different observations (e.g. flow measurements) can be used.

5.1.4 Comparison of EKF and EnKF

The main difference between the EKF and the EnKF is that the EKF is a one-shot procedure that uses only one state vector as estimation of the true state. The EnKF uses an ensemble of up to 100 state vectors: the average of the ensemble represents the estimated true state. This distinction has quite some impact on the algorithm: the EKF has to maintain both the average state $\boldsymbol{x}$ and covariance matrix P separately, where the EnKF estimates both quantities via the ensemble.

Reichle, Walker, Koster, and Houser (2002) describes five major differences between the EKF and the EnKF. The most important differences are here further explained and elaborated on.



(a) (b)

Figure 5.2: Example of updating using the EKF (a) and the EnKF (b). The red dot represents the prior (average) state, with if applicable in light red the different ensemble members. The green dot depicts the observation, and the yellow posterior state.

Algorithmic viewpoint

The EnKF estimates the ensemble in the prediction step via the non-linear model update. This implies that the prior state and the prior error covariance matrix are both non-linearly formed each time step by propagating a finite ensemble of model trajectories. This is in contrast with the EKF: the prior state is also non-linearly formed by the model, but the covariance matrix is linearly extrapolated from the posterior covariance matrix of the previous time step. The error covariance of the EnKF could thus be more accurate, but only if the ensemble is representative.

Both the EKF and the EnKF are based on linearisation. This can be seen by inspecting equations (5.5) and (5.6), which both say that the difference in state is linearly dependent on the difference between the observations and the predicted output by means of the matrix K. However, these linearisations are fundamentally different. The linearisation in the EKF is based on the Jacobian matrix (which represents the derivative) of the state and the observations around the estimated state. These derivatives can be found analytically, if that information is available, or using numerical approximations. The EKF is thus a *local* approach using only information at the estimated state, which is one point in the whole space of possible state vectors. The EnKF uses an ensemble of state vectors, and linearises using a linear fit between the ensemble members. The linearisation of the EnKF is thus essentially *non-local*: information outside the location of the (average) state in the state space is used in the linearisation. Concluding, the EKF is more an approach that *extrapolates* from the current state, and the EnKF essentially *interpolates* between the states.

One way this difference becomes visible is in the example of the "wrong sign" updating by the EKF (in a Eulerian coordinate system) (Yuan, 2013). See for example figure 5.2, where a (q, k) fundamental diagram as observation function is given. The current state is denoted with the red dot, and an observation is given by the green dot. In figure 5.2a the EKF procedure is depicted: the linearisation of the flow q with respect to the density

k is given by the red line. This red line has a positive slope, which results in a negative update of k due to an observation with a lower flow q. This update of k has the wrong sign: in reality the density k should be increased in order to match the observation. In figure 5.2b the same situation is given with some ensemble members given in light red. Here the linear fit of the ensemble has a negative slope, which leads to a correction with the correct sign. The EnKF seems less prone to large errors due to non-linearity than the EKF, assumed that the ensemble members are spread correctly.

Technical viewpoint

From a computational perspective, the fact that the EnKF represents the whole distribution by the ensemble members is quite advantageous. In principle, in the EnKF both the average state $\boldsymbol{x}$ as the covariance matrix P are defined by the state matrix X, which is a $n \times N$-matrix. Here denotes n the state size and N the ensemble size. The total number of stored elements is thus nN. For the EKF, one has to save a state vector $\boldsymbol{x}$ of size n and a covariance matrix P of size $n \times n$. As P is symmetric, one only needs to save e.g. the upper half of the matrix. The total number of elements saved is thus $\frac{1}{2}n(n+1)+n = \frac{1}{2}n(n+3)$. If we take an example of $n = 5000$, $N = 100$ and one element in the matrix is represented by a double-precision floating point number of 8 bytes, storing the needed elements for the EKF takes 100.06 MB instead of 4 MB for the EnKF with no overhead considered.

Moreover, the EnKF is easier to implement than the EKF. This is due to the fact that no derivative needs to be calculated analytically or numerically. Therefore the model part and the assimilation part can be separated. However, this argument only holds when the model and data assimilation can be separated by the simulation tool or programming language. If one for example can't (easily) initialize and run multiple instances of the simulation model separately, an ensemble-based approach is hard to manage.

In terms of computation time, the choice between the EKF and the EnKF depends on the used traffic model. A traffic model that takes long time to compute has more influence on the computation time of the EnKF algorithm, as this traffic model is run up to 50 times concurrently. However, the data assimilation algorithm of the EnKF is most likely more efficient: no derivatives need to be analytically or numerically calculated, and the covariance matrix P doesn't need to be updated or calculated explicitly. The exact computation time depends heavily on which algorithm is used. For example, only the localized EKF instead of the (standard) global EKF is suitable for large network estimation faster than real-time (Van Hinsbergen et al., 2012). For the EnKF, multiple reformulations are possible (e.g. using localization or using the Sherman-Morrison-Woodbury formula) to speed up the computation time; see subsection 5.2.1 and further for these reformulations.

5.1.5 Conclusion: the choice of the EnKF as preferred method

As described above, the Ensemble Kalman Filter has theoretical benefits over the Extended Kalman filter in some performance and algorithmic aspects. However, the ques-

tion remains if the EnKF is a feasible choice in terms of computation speed when applied to a large scale traffic model. The memory usage of the EnKF may be lower, but if the prediction of a large ensemble of traffic models takes a long time, the application of the EnKF is not feasible.

Almost no application of the EnKF in traffic engineering literature was found. Possible reasons for ignoring the EnKF could be a long computation time, but also unfamiliarity of the researchers with the EnKF. The exact reason the EnKF isn't the preferred choice is not clear.

In this thesis, the choice is made for the EnKF as the data assimilation method in this prototype, as the EnKF best fits in the modular architecture described in previous chapters. A simulation study should check if the EnKF could provide the expected accuracy within a reasonable time frame. In the next sections the EnKF if further theoretically analysed and extended in order to identify some good practices and pitfalls so the subsequent implementation performs well.

5.2 Theoretical analysis of the Ensemble Kalman Filter

In this section, the traditional EnKF as described in algorithm 3 is further analysed and adapted where needed.

5.2.1 Reformulation of EnKF equations for efficient computation

By rewriting the Kalman gain equation, it is shown that the observation matrix H is only needed as part of the matrix product HX (and thereby HA): (Mandel, 2009)

$$K_k = P_k^- H_k^\intercal \left(H_k P_k^- H_k^\intercal + R_{k-1}\right)^{-1} \tag{5.7}$$

$$= \left(\frac{AA^\intercal}{N-1}\right) H_k^\intercal \left(H_k \left(\frac{AA^\intercal}{N-1}\right) H_k^\intercal + R_{k-1}\right)^{-1} \tag{5.8}$$

$$= \frac{1}{N-1} AA^\intercal H_k^\intercal \left(\frac{1}{N-1} H_k AA^\intercal H_k^\intercal + R_{k-1}\right)^{-1} \tag{5.9}$$

$$= \frac{1}{N-1} A(H_k A)^\intercal \left(\frac{1}{N-1} (H_k A)(H_k A)^\intercal + R_{k-1}\right)^{-1} \tag{5.10}$$

Therefore, the matrix H doesn't need to be saved explicitly, but the matrix HX can be directly formed using the observation function $\boldsymbol{h}(\boldsymbol{x})$ can be on each ensemble member. This has some advantages:

- Creating the observation matrix H is often much harder than programming the observation function itself. (Mandel, 2006)

- The matrix H will in a typical application be sparse, e.g. when in the traffic engineering context a measurement links to only one cell. So a lot of elements in the matrix H would be zero. Saving the matrix H as a normal matrix can use a lot of memory, or additional effort has to be made to store the matrix as a sparse matrix. (Mandel, 2006)
- The direct use of the matrix HX makes it easier to include non-linear and smoothed observations from the model. The model can directly output the matrix HX without defining which observation function $\boldsymbol{h}$ is used and which state elements influences the data. This is especially important for our context, as the measurements are mostly non-linear (due to the fundamental diagram) and smoothed (due to the use of (e.g. 1 minute) time-smoothed measurements). Otherwise, one would need to redefine the matrix H for different values of X in order to correctly form HX.
- The "hidden" use of the observation function in the model will ensure the black-box characteristics of the data assimilation. Changes in for example the observation function (i.e. the fundamental diagram) implies no change in the working of the data assimilation algorithm, which ensures the separation between model and data assimilation described earlier.

5.2.2 Ensemble size and filter divergence

One of the main problems in EnKF approaches is the handling of the ensemble members. As in all sampling approaches, the trade-off exists between the accuracy of a large ensemble and the computational costs of a small ensemble. According to Oke, Sakov, and Corney (2007), it is essential that the ensemble adequately spans the model sub-space. This can be seen by expressing by recognizing that the changes to the mean model state by the data assimilation can be expressed, using equation (5.14), as:

$$\overline{\boldsymbol{x}_a} - \overline{\boldsymbol{x}_b} = Ac, \tag{5.11}$$

with c a N-dimensional column vector. This means that the update of the mean state is essentially a weighted sum of the ensemble members. Therefore the importance of the representativeness of the ensemble members is clear: if the ensemble doesn't span the same space as the forecast errors, no vector c can be found such the true state is reached (Oke et al., 2007). As an example: suppose $A = \begin{pmatrix} -1 & 2 & -1 \\ -2 & 4 & -2 \\ -3 & 6 & -3 \end{pmatrix}$, then no c can be found so that $\Delta\boldsymbol{x} = (1, 0, 0)^\intercal$; the mean state can only be updated in the direction of $(1, 2, 3)^\intercal$. The rank of A is thus crucial: it defines the linear subspace of the mean update. The rank of A is at maximum $N - 1$ (assuming that $N \leq n$), as one degree of freedom is removed by the fact that all rows sum to 0.

When the ensemble doesn't span the model subspace adequately, it is called *filter divergence*. A limited ensemble size increases the risk of filter divergence. As the system covariance is estimated by the sample covariance of the ensemble, an ensemble that is not spread enough will underestimate the process covariance and therefore put too much

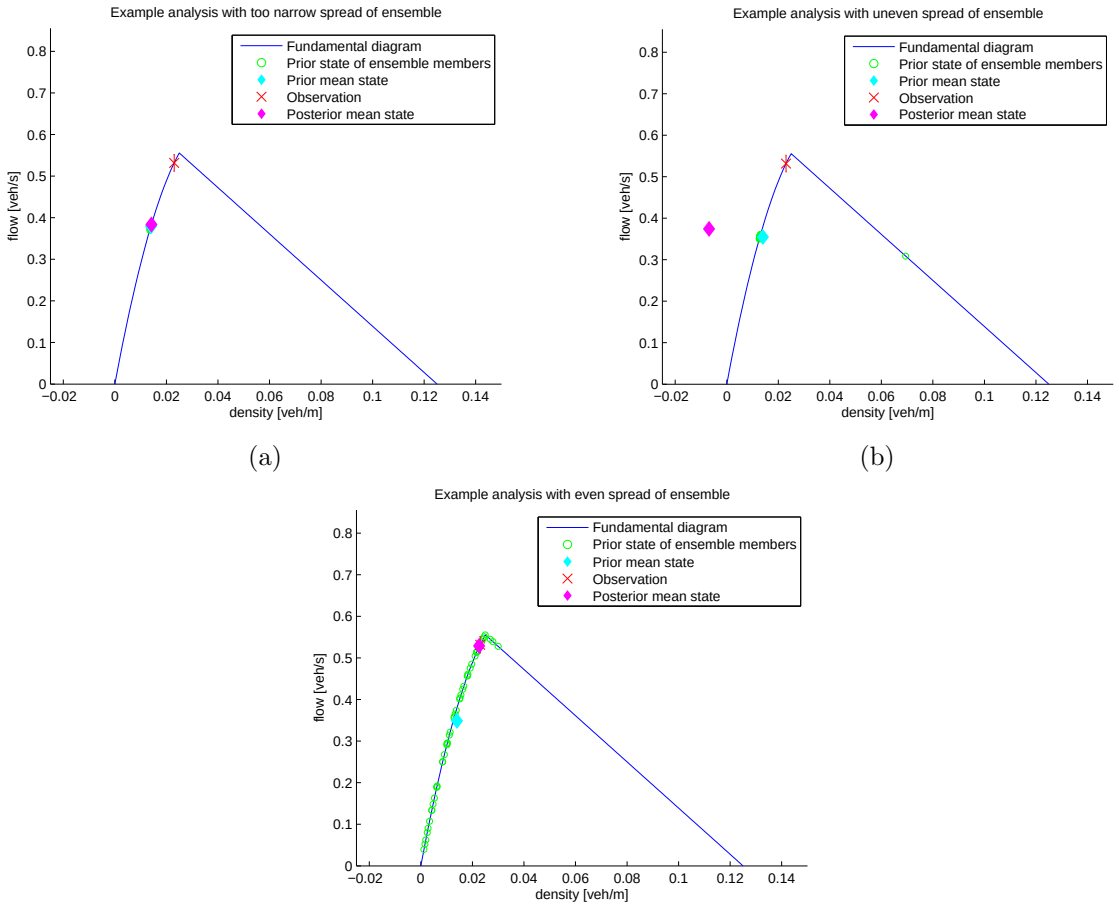


Figure 5.3: An example of an EnKF update step using only 1 state element (the density of a cell) and a non-linear observation function (fundamental diagram). Figure 5.3a depicts a situation where the ensemble is too narrowly spread, which causes the system to be unresponsive to a new observation; Figure 5.3b depicts a situation where the ensemble is unevenly spread, which leads to wrongly updating the state to a lower density instead of a higher density; Figure 5.3c depicts a situation where the ensemble is evenly spread. Note that the mean and the variance of the state in figure 5.3c is the same as in figure 5.3b.

confidence in the model. A filter that systematically underestimates the uncertainty of the process will lead to converging ensemble members and the filter becoming unable to adequately respond to new observations. See figure 5.3a for an example of a converged ensemble.

Another type of filter divergence is when some ensemble members converge to each other, but the covariances are maintained. However the values of the mean state and the covariance matrix are near the true values, there are no ensemble members that are near the true state. This leads to a decrease in effective ensemble size (as some ensembles describe the same situation and thus add no value to the analysis) and a high risk of an incorrect analysis, especially when the model is non-linear. In figure 5.3b and figure 5.3c the impact of such a uneven spread of the ensemble is visualized.

Another consequence of a limited ensemble size is the creation of spurious correlations (fake correlations). Spurious correlations are correlations between elements of the state that are not physically related and are at a significant physical distance from each other. These spurious correlations can lead to wrong updating of the state, as a measurement at a certain location will update the state at a wrong place. This is a direct effect of a limited ensemble size: the dimension of the system state is simply larger than the dimension of the ensemble.

In literature, numerous ways to diminish the risk of filter divergence are identified.

1. **Sampling strategies:** in literature several different sampling strategies are suggested. For example Houtekamer and Mitchell (1998) suggest separating the ensemble in two groups. The covariance of the one group is used in the computation of the Kalman gain for the update of the other group, and v.v. A problem that can arise in such sampling strategies is the inaccuracy of the Kalman gain computation as the ensemble size for computing the Kalman gain is essentially halved.

2. **Covariance inflation**: it is suggested that by artificially inflating the covariance (or equivalently the spread of the ensemble) at each update step, it is prevented that the covariance becomes too small. There are some different inflating strategies: multiplicative inflation of the state a priori or a posteriori or an additive inflation. In this research, it is important to acknowledge that the observation function is not well-defined (as it depends on previous states) so inflation of the prior state matrix X is not possible as the predicted observations matrix HX is then hard to update. The DEnKF has a implicit adaptive covariance inflation built in, as this method always overestimates the analysed error covariance. (Sakov & Oke, 2008)

3. **Localization**: in literature was found that localization is recommended (or even necessary) for large-scale EnKF approaches (Sakov & Bertino, 2011). The different localization approaches are further discussed in the next section.

5.2.3 Non-linearity in process model

In this subsection the influence of non-linearity of the propagation model is investigated. In the (standard) Kalman Filter, it is assumed that the propagation model is linear. The

Ensemble Kalman Filter relaxes this assumption, although the results of the EnKF are not optimal in non-linear conditions. on $\boldsymbol{f}$ in forecast step at time k. In general, the non-linearity of the propagation model implies that $\overline{\boldsymbol{f}(X_{k-1})} \neq \boldsymbol{f}(\overline{X_{k-1}})$, or in words, the mean of the propagated ensemble is not equal to the propagated ensemble mean.

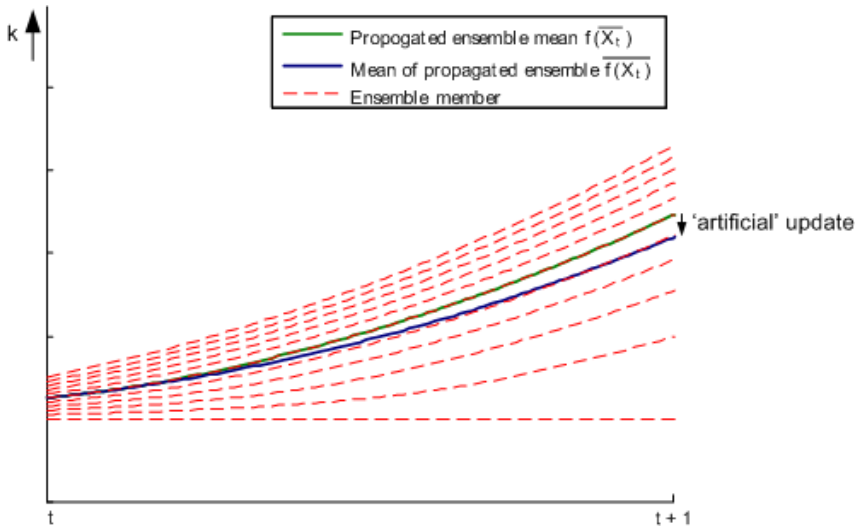


Figure 5.4: Difference between propagated ensemble mean and mean of propagated ensemble, and resulting artificial update

Consider for example an ensemble X_a of size $N = 11$ at time t (see figure 5.4). As the ensemble is nicely spread, the ensemble mean coincides with the 6th ensemble member (green line). However, at time $t + 1$, this 6th ensemble member lies above the ensemble mean at that point in time.

There are thus two main ways to represent the traffic state between the assimilation time steps. This intermediate traffic state is important as it is shown to the traffic operator and can be used for short-term predictions. One way is to represent the intermediate traffic state by propagating the ensemble mean ($\boldsymbol{f}(\overline{X_t})$, the green line in figure 5.4), or by the ensemble mean every model time step ($\overline{\boldsymbol{f}(X_t)}$, the blue line in figure 5.4).

The main advantage of the first method is that the intermediate traffic state are *physical* traffic states that comply with the traffic model at hand. Moreover, the first method is computationally better as the mean of the propagated ensemble doesn't need to be computed every model time step.

The main disadvantage of propagation of the ensemble mean to mean of the propagated ensemble is that the first method induces an *artificial* update. This artificial update occurs when for example at time $t + 1$ the innovation $\boldsymbol{d} - H\overline{\boldsymbol{x}} = 0$, the posterior state should be the same as the prior state. However, the traffic state is updated from $\boldsymbol{f}(\overline{X_{t+1}})$ to $\overline{\boldsymbol{f}(X_{t+1})}$, and thus a change in mean state occurs (see figure 5.4).

This size of this artificial update effect is influenced by a number of factors. This effect will be increased when the ensemble spread is large, the propagation model $\boldsymbol{f}$ is non-linear (diverging) and the assimilation interval is large.

5.3 Refinements to EnKF

In this section three refinements to the traditional EnKF are proposed. These three refinements form the basis of the tested assimilation methods in the experiments in the next chapter.

5.3.1 Sherman-Morrison-Woodbury formula

Matrix inversions are well-known for their computational costs. The computation of K_k according to the formula implies an inversion of a $m \times m$ matrix. If the number of observations m is large, the formulation above can be computationally expensive: the computational complexity of correction step of the traditional EnKF is $O(m^3 + m^2N + mN^2 + nN^2)$ (Mandel, 2006). This means that when the number of observations is doubled, the computation time can be multiplied by 8.

However, the inversion part of the Kalman gain equation can be reformulated using the Sherman-Morrison-Woodbury formula (Hager, 1989; Mandel, 2006):

$$W^{-1} = \left(R + \frac{1}{N-1}(HA)(HA)^\intercal\right)^{-1} \tag{5.12}$$

$$= R^{-1}\left[I - \frac{1}{N-1}(HA)\left(I + (HA)^\intercal R^{-1}\frac{1}{N-1}(HA)\right)^{-1}(HA)^\intercal R^{-1}\right]. \tag{5.13}$$

This SMW formula depends on the cheap inversion of the matrix R: as R in this context is mostly chosen diagonal, the computation of R^{-1} is easy. Note that equation 5.13 is a reformulation of 5.12, and not an approximation.

The SMW implementation reduces the total computational complexity to $O(N^3 + mN^2 + nN^2)$ (Mandel, 2006). See Appendix A for an in-depth analysis of the computational complexity.

5.3.2 Perturbation of observations: deterministic approaches

The EnKF approaches can be divided into two main classes: the stochastic approach that uses randomized observations for each ensemble member, and the deterministic approach that uses the same observations for the whole ensemble. As described in the algorithm above, the traditional EnKF is a clear example of a stochastic approach.

In this subsection the two stochastic and deterministic approach are further investigated. This is done by further investigating the update equation (5.6). The EnKF algorithm has two separate main functions: updating the mean state $\overline{\boldsymbol{x}_a}$ and updating the ensemble spread A_a. The update equation (5.6) can be split into a separate update of the mean of

the ensemble and the update of the deviations from the mean.

$$\overline{\boldsymbol{x}_a} = \bar{\boldsymbol{x}} + K\left(\boldsymbol{d} - H\bar{\boldsymbol{x}}\right) \tag{5.14}$$
$$A_a = A + K_A\left(D - HA\right) \tag{5.15}$$

The traditional stochastic approach consists of sampling D using the probability distribution of the measurement error R and using $K = K_A$. The deterministic approach sets $D = 0$ and $K \neq K_A$.

Traditional stochastic approach: the EnKF

The reason for using randomized observations is to ensure the right posterior error covariance. This can be made clear by explicitly calculating the posterior error covariance P_a: (Sakov & Oke, 2008)

$$P_a = \frac{1}{N-1} A_a A_a^\intercal \tag{5.16}$$
$$= \frac{1}{N-1}\left[A + K(D - HA)\right]\left[A + K(D - HA)\right]^\intercal \tag{5.17}$$
$$= P - PH^\intercal K^\intercal - KHP + KHPH^\intercal K^\intercal + \frac{1}{N-1} KDD^\intercal K^\intercal + \frac{1}{N-1}(I - KH)AD^\intercal K^\intercal + \frac{1}{N-1} KDA^\intercal (I - H^\intercal K^\intercal). \tag{5.18}$$

The last step is done by expanding the product and substituting $P = \frac{1}{N-1} AA^\intercal$. When the observations are not randomized (i.e. $D = 0$), the equation for the posterior error covariance becomes:

$$P_a = P - PH^\intercal K^\intercal - KHP + KHPH^\intercal K^\intercal \tag{5.19}$$
$$= (I - KH)P(I - H^\intercal K^\intercal), \tag{5.20}$$

which is smaller than the value proposed by the traditional Kalman Filter $P_a = (I - KH)P$. Without perturbing the data, this analysis scheme thus reduces the ensemble spread too much. (Burgers, Van Leeuwen, & Evensen, 1998)

The traditional way to solve this problem is to randomize the data, so that the $\frac{1}{N-1} DD^\intercal = R$. For this choice, the analysed P_a approximates the theoretical value: $P_a = (I - KH)P + O(N^{-\frac{1}{2}})$. (Burgers et al., 1998)

Deterministic approaches: the EnSRF and DEnKF

The other class of ensemble based Kalman filter approaches are the deterministic filters, also called the ensemble square-root filters (ESRF). Some examples of ESRF approaches are the EnSRF (Whitaker & Hamill, 2002), SEIK (Pham, 2001), EAKF (Anderson, 2001) and ETKF (Bishop, Etherton, & Majumdar, 2001). See Tippett, Anderson, Bishop,

Hamill, and Whitaker (2003) for a further discussion of these variants. Here the EnSRF is further elaborated on.

In the traditional EnKF, $K_x = K_A$ and D is sampled using the probability distribution of the observations R. In the EnSRF, $D = 0$ in equation (5.15) as the observations are not perturbed. Equation (5.15) thus simplifies to $A_a = A - K_A HA = (I - K_A H)\, A$. In order to fit K_A that the analysed ensemble will have the right covariance matrix, it can be shown that K_A can be chosen as: (Andrews, 1968)

$$K_A = P^b H^\intercal \left[\left(\sqrt{HP^b H^\intercal + R}\right)^{-1}\right] \times \left[\left(\sqrt{HP^b H^\intercal + R} + \sqrt{R}\right)^{-1}\right] \tag{5.21}$$

The calculation of K_A thus requires the computation of square roots of $m \times m$ matrices. This can be done using Cholesky factorization or singular value decomposition, but requires accurate linear algebra packages and imposes significant computational overhead. Note that when measurements are updated one at a time, the square root and inverse operations are operated on scalars which reduces the computational complexity significantly. Moreover, Leeuwenburgh, Evensen, and Bertino (2005) suggest that using ESRF approaches in nonlinear dynamics, all members but one tend to collapse into one state instead of a nice Gaussian ensemble spread. Moreover, the application of localization techniques (see next paragraph) is harder using a ESRF approach.

Sakov and Oke (2008) describe the DEnKF (deterministic EnKF), which is a hybrid approach that "combines the performance of the ESRF and the simplicity and versatility of the EnKF" (Sakov & Oke, 2008, p. 370). As the ESRF, it doesn't rely on perturbation of the observations. However, it approximates the theoretical posterior error covariance. In essence, the basis of the DEnKF is choosing $K_A = \frac{1}{2}K$.

The posterior error covariance without perturbation of observations (see equation (5.19)) is analysed for this choice. As $PH^\intercal K^\intercal = KHP$ (which can be proved by substituting K), equation (5.19) can be written as:

$$\begin{aligned} P_a &= P - PH^\intercal K_A{}^\intercal - K_A HP + K_A HPH^\intercal K_A{}^\intercal && (5.22)\\ &= P - \frac{1}{2}PH^\intercal K^\intercal - \frac{1}{2}KHP + \frac{1}{4}KHPH^\intercal K^\intercal && (5.23)\\ &= P - KHP + \frac{1}{4}KHPH^\intercal K^\intercal && (5.24)\\ &\approx (I - KH)P && (5.25) \end{aligned}$$

When KH is small in some sense, and thus the quadratic term is very small, the posterior error covariance matches the theoretical error covariance.

The question remains if the approximation is good in the context of this application. Sakov and Oke (2008) had some promising results in the application of the DEnKF in some (small) applications.

The DEnKF algorithm thus consists of roughly the following steps:

1. Calculate the mean posterior state $\overline{\boldsymbol{x}_a}$ using equation (5.14).

	Traditional EnKF	EnSRF	DEnKF
Observation perturbation	Yes	No	No
Separation between update of mean and update of ensemble spread	No, $K = K_A$	Yes, independent K and K_A	Yes, linear dependent: $K_A = \frac{1}{2}K$
Match posterior error covariance with theoretical values	Approximation, statistical	Exact	Approximation, analytical
Implementation	Very Easy	Hard	Easy
Adding refinements, e.g. localization	Very Easy	Hard	Easy

Table 5.1: Overview characteristics of the EnKF, EnSRF and DEnKF approaches

2. Calculate the posterior ensemble anomalies using

$$A^a = A^f - \frac{1}{2}KHA^f.$$

3. Calculate the full posterior state by $X_a = A_a + [\overline{\boldsymbol{x}_a}, \ldots, \overline{\boldsymbol{x}_a}]$.

This approach is equivalent to updating the whole ensemble using equation (5.6) with half the Kalman gain and no perturbed observations, and then shifting the ensemble mean to the explicitly calculated ensemble mean by equation (5.14).

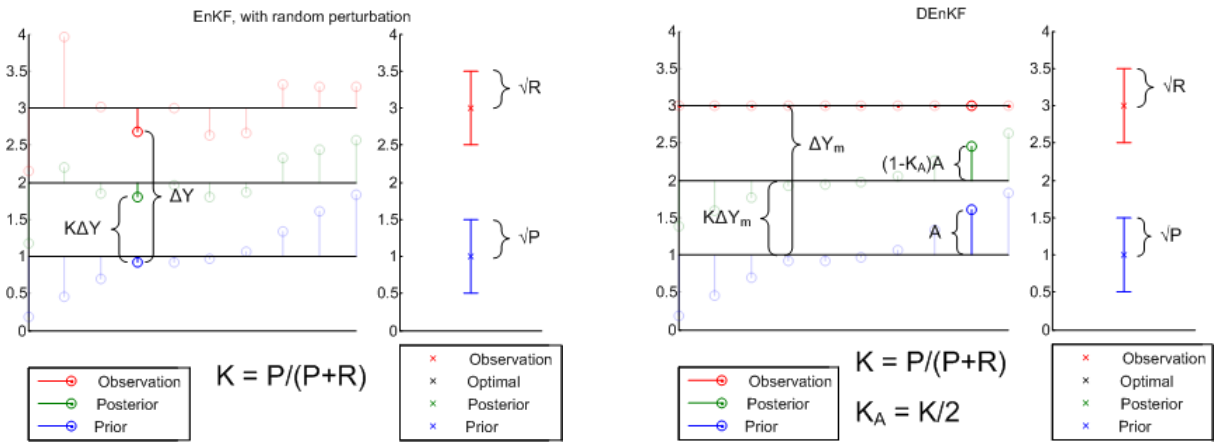
Comparison between stochastic and deterministic approaches

In table 5.1 the characteristics of the stochastic and the two deterministic approaches are summarized.

These deterministic approaches have as advantage that they in contrast to the traditional EnKF don't depend on the randomly chosen realization of the observations. The deterministic nature makes verification easier, as results stay the same in different runs. The deterministic nature also reflects in the ease-of-use when the whole data assimilation tool is used in practice. When for example an evaluation is made in which scenarios the data assimilation tool performs badly, it is useful when the performance can be accounted to the specific scenario instead of some randomness in the data assimilation.

Example of differences between EnKF and DEnKF

In this subsection the difference between the traditional EnKF and the DEnKF is graphically explained.



(a) (b)

Figure 5.5: Example of the update of an ensemble member in the traditional EnKF algorithm (figure (a)) and the DEnKF algorithm (figure (b))

An simple situation is chosen with only 1 state element, which is directly observed (which implies $H = 1$). An ensemble size of 10 is chosen. The prior state is normal distributed with mean $\overline{\boldsymbol{x}} = 1$ and variance $P = \frac{1}{4}$. An observation is measured with value $\boldsymbol{d} = 3$ and measurement error $R = \frac{1}{4}$. When the normal Kalman Filter is used, the Kalman gain K can be computed by $K = PH^{\intercal}\left(HPH^{\intercal} + R\right)^{-1} = \frac{1}{2}$. The mean posterior state thus is derived as (5.14): $\overline{\boldsymbol{x}_a} = \overline{\boldsymbol{x}} + K\left(\boldsymbol{d} - H\overline{\boldsymbol{x}}\right) = 2$, with $P_a = (I - KH)P = \frac{1}{8}$. The ensemble based Kalman Filter should approximate these values in this linear example.

Firstly the basic update procedure of the EnKF and the DEnKF are graphically explained. After that, the reason for using the DEnKF, i.e. the consistent estimation of the posterior state error, is graphically explained.

Update using EnKF and DEnKF Figure 5.5 depicts the basic update procedure of the traditional EnKF and the DEnKF.

For the EnKF, the Kalman gain is derived using the ratio between the prior state error and the sum of the state and measurement error. Each ensemble member is independently updated using the Kalman gain on the difference between the observation and the predicted observation. The measurement is perturbed for each ensemble member.

For the DEnKF, firstly the mean update is determined using the mean prior state and the measurement. Secondly, the deviation of each ensemble member from the ensemble mean is reduced by a factor $1 - K_A$. The measurements are not perturbed.

Reason of using DEnKF In figure 5.6a the situation is depicted of using the EnKF without perturbing the observations. Every ensemble member is updated with half the difference between observation (indicated in red) and the prior state (indicated in blue), as the Kalman gain is equal to $\frac{1}{2}$. The updated ensemble is indicated in green. When the traditional EnKF is used without perturbing the observations for each ensemble member,

the posterior state covariance becomes too small as in equation (5.19). This is indicated in 5.6a, where the spread of the posterior state is too small in comparison with the optimal spread.

In order to statistically approximate the right posterior state covariance, in the traditional EnKF the observations are perturbed using the measurement error R. See figure 5.6b for this example. Also here, the posterior state is formed by the average distance between the observation and the prior state. The posterior state spread is quite close to the ideal value.

However, the accuracy of this statistical approximation is quite dependent on values of the samples measurement errors. This can be seen by inspecting figures 5.6c and 5.6d, which use the same measurement error sample, but sorted differently. In figure 5.6c the measurement error sample is sorted descending, which causes the state spread to be dampened. This leads to a very narrow posterior ensemble. In figure 5.6d the opposing scenario is depicted, in which the state spread is maintained as much as possible.

The spread of the ensemble is thus quite dependent on the values of the measurement error that are coincidentally sampled. Therefore the DEnKF is used, as depicted in 5.6e. The DEnKF update essentially consists of two steps. Firstly the mean update is computed. As the Kalman gain was computed to be $\frac{1}{2}$, the mean posterior state is halfway between the mean prior state and the observed value. Secondly, the posterior deviations around its mean are computed. For these deviations, another Kalman gain $K_A = \frac{1}{2}K = \frac{1}{4}$ is used. This means that the deviations of the posterior state are $\frac{3}{4}\times$ the deviations of the prior state. This approximation is quite good in this example.

The rule of thumb $K_A = \frac{1}{2}K$ used by the DEnKF is only an approximation. As in this simple situation using only one state element/observation, the optimal value K_A can be easily calculated exactly using (5.21). This exact calculation is done in the EnSRF approaches. The optimal K_A was found at $K_A = 0.2929$, see figure 5.6f.

5.3.3 Localization

As identified in the previous subsection, localization is one of the possible solutions to all kinds of problems of ensemble based Kalman Filters. Localization provides in essence extra information to the filter, namely the (geographical) network description. The basis of the localization techniques is deleting or minimizing the relations between model elements, i.e. state elements or measurements, that are physically distant. The assumption is made that these distant relations are not relevant in the physical world: the correlations between distant model elements are spurious.

In short, two main reasons exist for using localization in the EnKF: (Sakov & Bertino, 2011)

1. Deleting spurious correlations ('fake' correlations) between physically distant model elements (Hamill, Whitaker, & Snyder, 2001; Houtekamer & Mitchell, 2001). Spurious correlations are correlations that arise in the equations due the estimation of the covariance matrices by a limited ensemble size. These correlations govern the

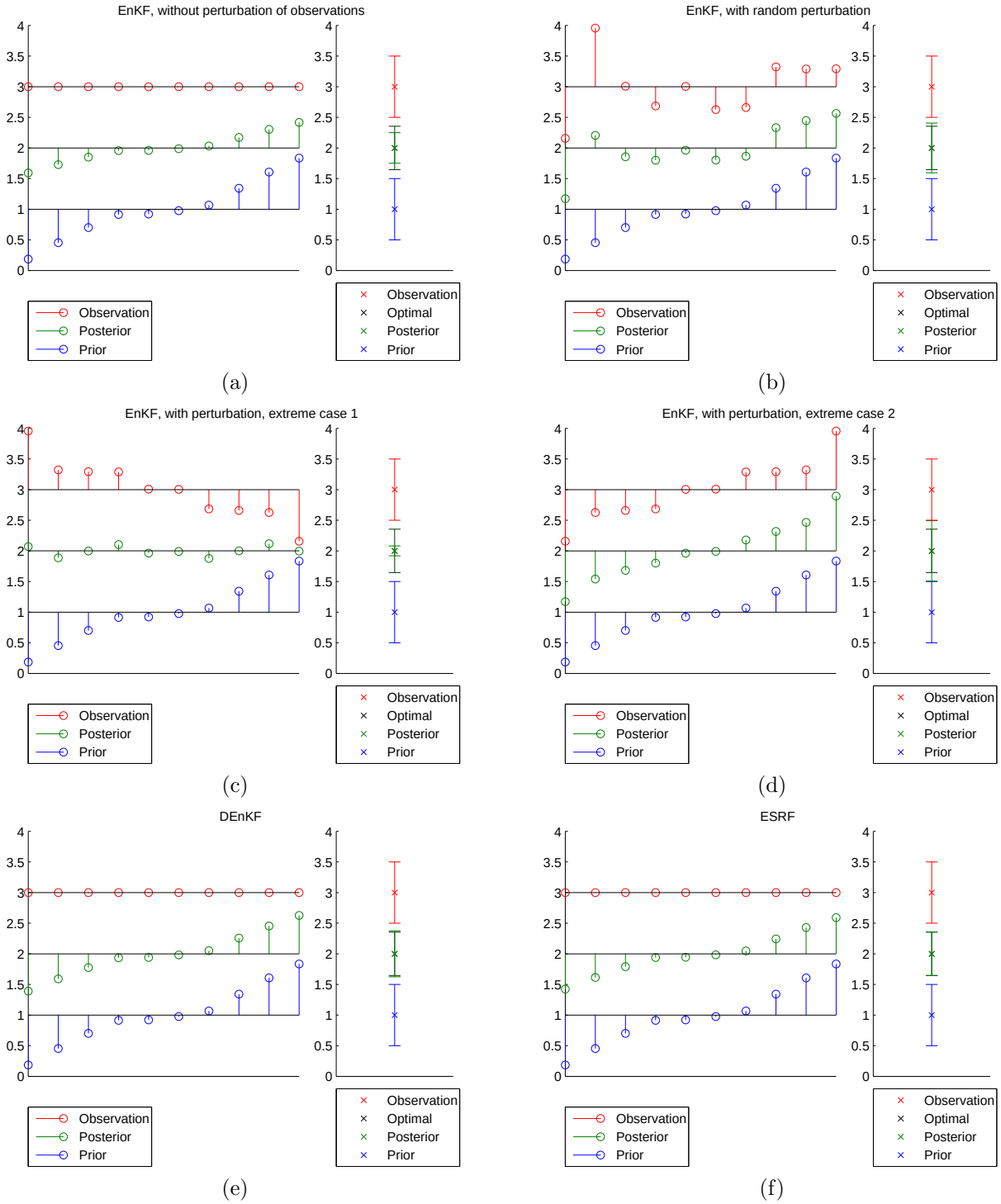


Figure 5.6: This figure indicates the results of six examples of the different assimilation schemes. Each subfigure consists of two subplots: the left subplot depicts the analysis of the ensemble members, where the prior state and the observation are combined into the posterior state. The right subplots depict the analysis of the mean state. The crosses indicate the mean value, and the bars indicate the associated error (standard deviation) of the variables.

impact of the observations on the state elements: a large correlation increases the sensitivity of a state element to the innovation of an observation (the difference of the predicted observation and observed value). Spurious correlations will thus correct the state in a wrong way.

2. Insufficient ensemble rank. In general the degrees of freedom of the ensemble is less than the degrees of freedom of the model. Localization increases the effective ensemble size by in essence decoupling the model into several independent parts, which are separately solved by the ensemble. (Oke et al., 2007)

The second reason is further developed in an example in figure 5.7. In figure 5.7a a situation is depicted where an ensemble consisting of three ensemble members. The x-axis depicts for example a road stretch, where the y-values indicate the density on location x. The dotted lines are the trajectories of the ensemble members, which combine to the green line. The red line indicates the truth, to which the ensemble is going to be fitted.

In figures 5.7b and 5.7c the way of working of the global method is depicted. In 5.7b the deviations to the ensemble mean are depicted. The blue line indicates the updated deviation of the mean, which is found by a linear sum of the ensemble members. In this example, the blue line is found by $\Delta x = A \cdot [-0.0384; -0.0720; -0.5725]$. In 5.7c it is shown that the updated posterior state fits the truth state much better than the prior state.

In figure 5.7d and 5.7e a local method is used. The state vector is split into three regions, which are individually updated. In this case, the blue line is found by

$$\Delta x = \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \end{pmatrix} \begin{matrix} A_1 \cdot [-0.1794, 0.0061, -0.3115]^\intercal \\ = A_2 \cdot [3.8815, 4.2998, 3.1438]^\intercal \\ A_3 \cdot [-0.1211, -0.1310, -1.1509]^\intercal \end{matrix}$$

For each region, another combination of ensembles can be chosen to fit the truth line. This is what is meant by increasing the effective ensemble size. The local method works as if a total of 9 ensemble members exist. The result of the localized method in figure 5.7e is much better than the result of the global method in figure 5.7c.

However the accuracy is better, the bounds of the regions can cause sudden discontinuities ('jumps') in state, which may not be possible in a global method. This is a major disadvantage of localization.

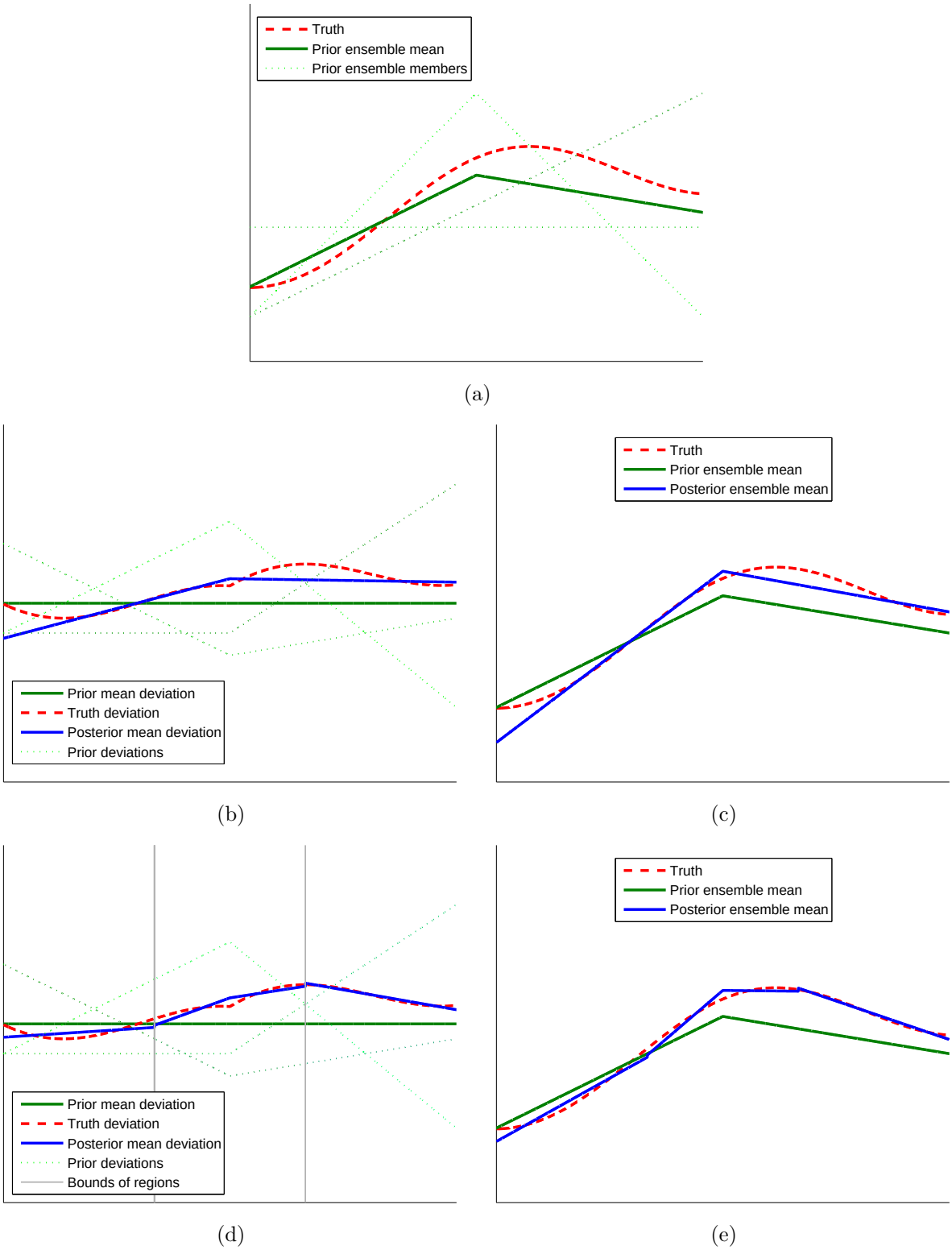


Figure 5.7: An example of a global and local analysis. The figures plot the values of the state vectors.

In this section three localization techniques are further elaborated on: covariance localization, state based local analysis and observation based local analysis.

Covariance localization

Covariance localization: covariance localization is a technique that restricts the update of elements of the state to measurements in the (physical) vicinity of that element. This is done by replacing the state error covariance P (in the EnKF estimated by its sample covariance C) by its element-wise product (also called the Schur product or the Hadamard product) with a distance-based function ρ: $P_{new} = \rho \odot P$. Sakov and Bertino (2011) has illustrated the method clearly, as can be seen in figure 5.8.

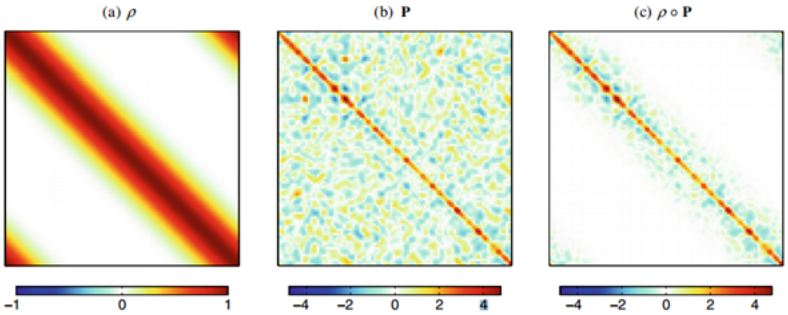


Figure 5.8: Covariance localization: filtering the state error covariance matrix P with a distance-based function ρ. From Sakov and Bertino (2011).

One has multiple possibilities for choosing the function ρ: simple functions as $\mathbb{1}(d_{ij} < \delta)$ which has value 1 for measurements close enough, and value 0 for measurements further away, or smoother functions that weigh the nearby measurements based on distance, e.g. a 5th-order piecewise rational function (Gaspari & Cohn, 1999). It is important to consider that ρ should be positive-definite, as then $\rho \odot P$ is also positive definite and thus the factor $\left(H_k P_k^- H_k^\intercal + R_{k-1}\right)$ in the computation of the Kalman gain is positive definite and thus invertible. For large systems, where P is not explicitly calculated (see subsection 5.2.1), it can be chosen to localize the Kalman gain directly: $K_{new} = \rho \odot K$.

Local analysis (state based)

Local analysis is a similar technique as covariance localization, based on the same principle of restricting the update of the state to measurements in the physical vicinity of the elements of the state. It approximates the state error covariance for each state vector element by solely considering a "virtual local spatial window around this element" (Sakov & Bertino, 2011). The difference can be seen in the figure 5.9 (from (Sakov & Bertino, 2011)) when compared to the figure above. In essence, the state is updated element by element using a subset of observations located near the currently updated element. (Evensen, 2003)

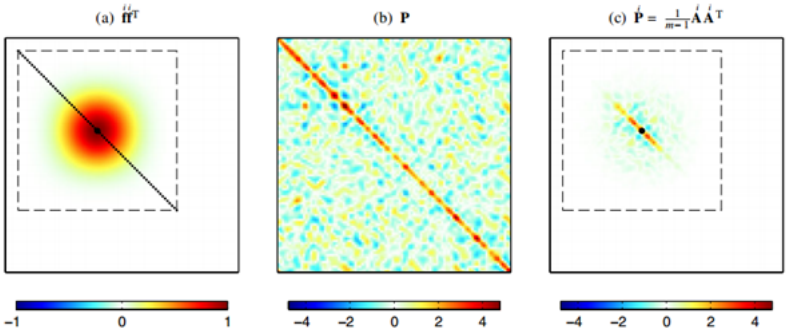


Figure 5.9: Local analysis: considering only a local window around a certain state element. From Sakov and Bertino (2011).

Algorithm 4: the (state based) localized Ensemble Kalman Filter

Consider a (non-linear) state space model as in equations (5.1) and (5.2), where $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are independent, zero-mean, Gaussian noise processes with covariance matrices Q_k and R_k. Consider the observation matrix H_k given for each k.
Initialization:

$$X_0 = \mathrm{E}\left[X_0\right]$$

for $k = 1, 2, \ldots$ **do**

Step 1a: predict global mean and variance of state variables (forecast step)

$$X_k^- = \boldsymbol{f}(X_{k-1}) = \left[\boldsymbol{f}(\boldsymbol{x}_{k-1}^1), \boldsymbol{f}(\boldsymbol{x}_{k-1}^2), \ldots, \boldsymbol{f}(\boldsymbol{x}_{k-1}^N)\right]$$

$$P_k^- = \frac{AA^\intercal}{N-1}$$

$\boldsymbol{x}_t^i$ denotes the i-th ensemble member at time t, and $A = X_k^- - \mathrm{E}\left[X_k^-\right]$ is the anomalies matrix
Step 1b: predict global output

$$Y_k^- = H_k X_k^-$$

Step 1c: perturb observations

$$D = [\boldsymbol{d} + \boldsymbol{\epsilon}_1, \boldsymbol{d} + \boldsymbol{\epsilon}_2, \ldots, \boldsymbol{d} + \boldsymbol{\epsilon}_N]$$

for *every state element x in the state vector* $\boldsymbol{x}$ **do**

Step 2a: select elements corresponding to that state element
Select row vector $\boldsymbol{x}_L$ from X_k^- corresponding to the state element x. Select the row vector $\boldsymbol{a}_L$ from A_k analogously.
Select corresponding observation matrix D_L, HX_L and HA_L from D, HX and HA respectively. Select also the corresponding matrix R_L from R.
Step 2a: compute Kalman gain

$$K_k = \frac{\boldsymbol{a}_L (HA)^\intercal{}_L}{(HA)_L (HA)^\intercal{}_L + R_L}$$

Step 2b: update mean and covariance

$$\hat{\boldsymbol{x}}_L = \boldsymbol{x}_L + K_k(D_L - HX_L)$$

Step 2c: update global matrix
Update X_k^- by replacing the corresponding cells with $\hat{\boldsymbol{x}}_L$. Moreover, compute $\hat{\boldsymbol{a}}_L$ and substitute into A_k.

end

end

The consequence of using this approach to the computational costs are significant. Using the straightforward implementation, the computational complexity reduces from $O(m^3 + m^2N + mN^2 + nN^2)$ to $O(m_l^3 n + m_l^2 nN + m_l nN^2)$ (see appendix A). As for large scale

applications where the number of measurements m is big in comparison to the localization radius m_l, this reduction of computational complexity is significant.

Another reason for using local analysis is that in large-scale applications the number of measurements m is typically larger than the ensemble size N. This can lead to singularity problems, i.e. the matrix $((HA)(HA)^\intercal)/(N-1)+R)$ is not invertible. This problem can be solved by using a pseudo-inverse or low rank approximations (Evensen, 2004), but it is not ideal. Moreover, by transforming the updating from a (n, m, N) problem to a $(1, m', N)$ problem, the model is solved in a relatively large ensemble space. The local analysis scheme therefore "significantly reduces the impact of a limited ensemble size and allows for the use of EnKF with high-dimensional model systems" (Evensen, 2009, p. 101).

Local analysis (observation based)

Observation based local analysis is similar to state based local analysis. Instead of iterating over each state element, only one measurement is selected every iteration. Suppose that for every observation only n' state elements are selected. This way, the computational complexity reduces to $O(mn_lN^2 + nN)$ (see appendix A) . Note that now the inverted matrix is reduced to a scalar. Moreover, when $m < n$, the update equation has to be iterated less than in the state based local analysis. So this method has computational benefits over the state based local analysis.

However, in contrast to the state based local analysis, the matrix with measurements HX also needs to be updated after an update of the state X. This might pose a problem, as the observation function is not readily available (due to invisibility for the data assimilation package in a black box approach). Therefore, two possible solutions exist:

- Force calculation of an approximation of the synthetic observations by the model. This can for example be the calculation of instantaneous observations (by applying the fundamental diagram) instead of using the time-smoothed values.
- Approximate observations by the use of the ensemble. Consider that the original vectors $\boldsymbol{x}_0$ (state) and $\boldsymbol{y}_0$ (observations) are given. Now, given the change of the state to $\boldsymbol{x}_a$, the value of $\boldsymbol{y}_a$ needs to be found. This can be solved by using (linear) regression. If it is assumed that the value of an observation is only dependent on one state element (e.g. the speed observation at a location is dependent on the density at that location), β_1 is a scalar in the following linear regression equation:

  $$\boldsymbol{y}_a = \boldsymbol{y}_0 + \beta_1 (\boldsymbol{x}_a - \boldsymbol{x}_0) .$$

  β_1 is obtained using linear regression, i.e.: $\beta = \binom{\beta_1}{\beta_2} = (HH^\intercal)^{-1}HY$ with $H = \binom{\boldsymbol{x}_0}{1}$ and $Y = \boldsymbol{y}_0$.

Comparison localization methods

Sakov and Bertino (2011) compares the covariance localization and local analysis techniques. It concludes that the two techniques are formally different, in practice the two techniques should yield similar results.

The local analysis makes it easier to incorporate reformulations of the update equations, as the covariance localization is based on approximations which may not hold in e.g. non-linear conditions.

5.4 Conclusions

There are multiple changes or extensions to the traditional EnKF scheme that theoretically provide benefits for using the EnKF in estimating the state of a large-scale traffic model. The "hidden" observation function provides the use of non-linear and smoothed observations. Covariance inflation can save the EnKF filter from becoming unresponsive to new measurements. The use of a deterministic algorithm can reduce the impact of coincidental sampling and associated sampling errors. Most important is the inclusion of localization as it has several possible advantages for large-scale applications: blocking unnecessary correlations, decreasing computation time due to smaller matrix inversions, increasing the accuracy by increasing the effective ensemble size.

Chapter 6

Implementation of prototype

In this chapter the implementation of the prototype is described. The implementation is based on the literature review in chapter 4 and 5.

6.1 General information

In order to incorporate the modularity of the architecture, it is chosen to adopt an object-oriented approach and associated programming language.

The architecture is built as an extension to (an old version of) the OpenTrafficSim project, which is an initiative to combine multi-scale and multi-modal traffic simulations. This means that basic classes for visualization can be reused. The OpenTrafficSim project is programmed in Java.

The source code can be found on GitHub: `http://www.github.com/frisoscholten/OTSim-Macroscopic-DA`.

6.2 Traffic flow model

Initialization: All cells are joined based on same lanes, speed limit, fundamental diagram properties and no merges and diverges between cells. Then split according to defined minimum length (based on CFL).

Network: cell and node classes cell has density; Demand and supply calculated via FD interface . Node: calculates fluxes using demand and supply. Several FD classes can be set as implementation of FD interface. Every cell has begin and end nodes

Detector special case of node; stores velocity and flow of associated cell every time step. For memory reasons "old" measurements are removed.

In this prototype inflows can be set time dependent, turn fractions not, but that can be changed in later versions.

6.3 Input and output

In order to compute the performance, intermediate performance metrics are computed every time step. At the end of the analysed time window the intermediate performance metrics are combined into an overall performance. This procedure using intermediate performance is used due to memory constraints: otherwise the whole state over the whole time horizon must be stored.

6.4 Data assimilation

In order to incorporate prior knowledge of the inflow pattern, the inflow pattern is modeled as $I(t) = \gamma I_p(t)$, where $I_p(t)$ is the prior inflow pattern. In the data assimilation methods, the value of γ is estimated in order to fit the inflow $I(t)$ to the observations.

The time propagation of the ensemble members every assimilation time step is done in a multi-core setting, as the propagation of an ensemble members is independent from the other ensemble members. This multi-core propagation is done using the ForkJoin framework (Lea, 2000).

The data assimilation classes are implemented in a quite general way: one can easily decide which variables are present in the state vector (e.g. cell density, inflow and turn fractions, but also fundamental diagram parameters).

6.4.1 Matrix implementation

Matrix is based on JAMA class due to the ease of implementation and integrated methods as the Cholesky decomposition. Disadvantages are possibly the absence of sparse matrices and the non-optimal speed of matrix operations.

In order to find the best (matrix) implementation that can do the update equations the fastest, some small tests were done. A test class was built, where the matrices X, A, HX, HA, R and D were filled with random values and the size of the matrices were configurable. The computations of the following equation was timed:

$$\begin{aligned} \Delta X &= K(D - HX), \\ \Delta X &= A(HA)^\intercal \left((HA)(HA)^\intercal + R\right)^{-1} (D - HX). \end{aligned} \tag{6.1}$$

Equation (6.1) can be solved in many ways. In general, two main considerations are to be made: firstly the choice if the Kalman gain K is explicitly or implicitly calculated, and secondly the choice that direct solving is used or solving via the Cholesky decomposition. This leads to four different methods. The four methods are as follows:

Solving via Cholesky decomposition, with (on the left, method 1) implicit and (on the

right, method 2) explicit Kalman gain computation:

$$\begin{aligned} P &= ((HA)(HA)^\intercal + R) \\ L &= \text{Chol}(P) \\ M &= \text{Solve}(L, D - HX) \\ \Delta X &= A(HA)^\intercal M \end{aligned} \qquad \begin{aligned} P &= ((HA)(HA)^\intercal + R) \\ L &= \text{Chol}(P) \\ M &= \text{Solve}(L, I_{m\times m}) \\ K &= A(HA)^\intercal M \\ \Delta X &= K(D - HX) \end{aligned}$$

Directly solving, with (on the left, method 3) implicit and (on the right, method 4) explicit Kalman gain computation:

$$\begin{aligned} P &= ((HA)(HA)^\intercal + R) \\ M &= \text{Solve}(P, D - HX) \\ \Delta X &= A(HA)^\intercal M \end{aligned} \qquad \begin{aligned} P &= ((HA)(HA)^\intercal + R) \\ M &= \text{Solve}(P, I_{m\times m}) \\ K &= A(HA)^\intercal M \\ \Delta X &= K(D - HX) \end{aligned}$$

The computations are done 10 times with $n = 4000$, $m = 1000$ and $N = 20$, which are similar conditions as in the Rotterdam highway network case. The computation times are give in table 6.1. According to this small test, the Cholesky decomposition with an implicit calculation of the Kalman gain is the best.

Implementation	Mean computation time [ms]	Standard error [ms]
1: Cholesky, implicit K	394	15.4
2: Cholesky, explicit K	17789	48.1
3: Direct, implicit K	489	12.3
4: Direct, explicit K	1398	12.7

Table 6.1: Mean computation times with associated standard errors for the four different implementations.

For the DEnKF, in essence two equations have be solved:

$$\Delta X_1 = A(HA)^\intercal \left((HA)\,(HA)^\intercal + R\right)^{-1} \left(\overline{D} - \overline{HX}\right)$$

$$\Delta X_2 = A(HA)^\intercal \left((HA)\,(HA)^\intercal + R\right)^{-1} (D - HX)\,,$$

in which the overbar indicates the mean over the ensembles.

Therefore it has to be decided if it's faster to explicitly calculate K, as K is the common factor in the two equations, or it's faster to solve the equations twice. Table 6.1 indicates that it will be faster to solve the equations twice, as the explicit K methods are at least 3.5 times as slow as the implicit methods.

6.5 Verification of prototype

In this section the implementation of the prototype is verified. The goal of the verification is to check if any errors in implementation occurred. A succesful verification

makes sure that the prototype behaves as intended and builds confidence in the results of the prototype. Note that in contrast to validation, the verification only checks if the implementation behaves as the modeller intends to. In validation the link between the behaviour of the model and the real world behaviour is checked.

This section is divided into a few subsections. In the first subsection the implementation of the traffic flow model is verified. The second subsection checks the implementation of the data assimilation method.

6.5.1 Verification of traffic flow model

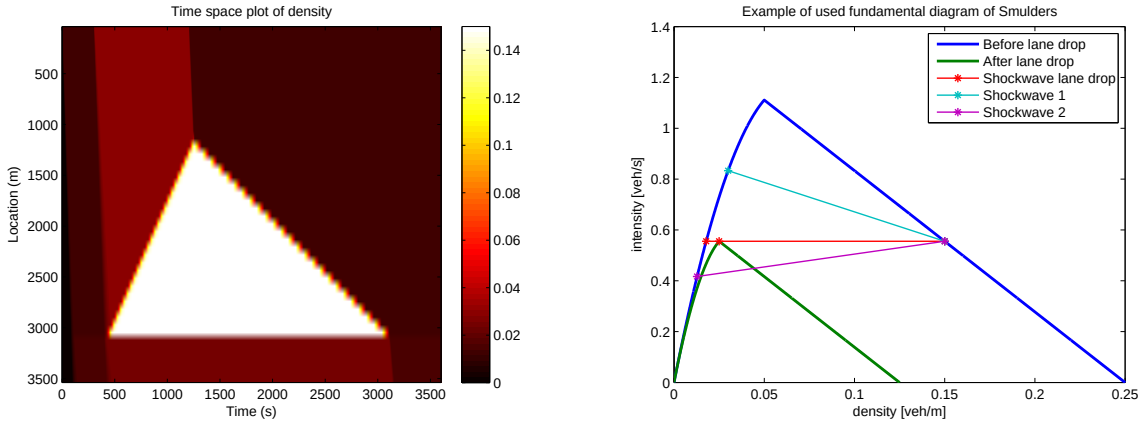
In this subsection the traffic flow model is verified. The following list indicates some of the verification tests done, both by checking automatically and manually.

- ✓ Unit tests cell methods, e.g. update of the density .
- ✓ Unit tests node methods, e.g. computation of the fluxes.
- ✓ Check if cells have correct length, i.e. not too small for the CFL condition or unreasonable large.
- ✓ Check if cells have physical attributes, e.g. density and flows shouldn't be negative.
- ✓ Check if fundamental diagram is used correctly by plotting all speeds and flows corresponding to the densities.

The verification tests above only check the traffic flow model implementation on a certain sublevel: e.g. only at one time step or at one submodel. In order to verify the implementation of the traffic flow model as a whole, a simple case is considered of a highway stretch with a lane drop. When considering different fundamental diagrams and inflow patterns, the resulting congestion patterns can be checked by shockwave theory, as the first order traffic flow model should model these shockwaves.

Several tests are done using different inflow patterns and fundamental diagram parameters. In all tests, the shockwave speeds are identified using two methods: a computation using the model output and a computation using only the fundamental diagram. An example of a test is displayed in figure 6.1. In all tests, the shockwave speeds using the two methods only differ slightly (1% at most). These differences can be explained by the error imposed by discretizing the model, as the differences in speed (ϵ_v) are smaller than a cell length (L_{cell}) in the time interval (Δt) the shockwave exists: $\epsilon_v \Delta t < L_{cell}$.

As the implementation of the traffic flow model passed all the previous checks with success, it is concluded that traffic flow model is correctly implemented.



(a) (b)

Figure 6.1: An example of the results of the traffic flow model in a lane drop case. The lane drop occurs at location $x = 3000$. The inflow pattern is 1500 veh/h with a peak of 3000 veh/h between $t = 300$ and $t = 1200$. Three shockwaves exist: a horizontal shockwave at $x = 3000$, shockwave 1 between the high density region and the traffic jam and shockwave 2 between the low density region and the traffic jam. Figure (a) describes the time space plot of the density of the cells. The shockwaves surround the triangle: at the bottom border the horizontal shockwave, shockwave 1 at the left border and shockwave 2 at the right border; Figure (b) describes the used fundamental diagram with the calculation of the shockwaves.

6.5.2 Verification of data assimilation

- ✓ As the data assimilation is matrix based, errors would occur if matrices having incompatible sizes are added or multiplied. Automatic checks are provided to ensure that the matrices have the correct size. In this way it is ensured that the equations are correct.
- ✓ Some methods, for example the noise generation method in order to perturb vectors, are tested independently.
- ✓ The correction equations of all the data assimilation methods are manually checked using synthetic input.
- ✓ For some model runs it is manually checked that the model states of the ensemble members are correctly put into the corresponding matrices.
- ✓ For some model runs, it is manually checked that the model states are correctly updated with the values of the (updated) model state matrix.

Assimilation of equivalent assimilation methods

In this research, two different implementations of the correction step of the ensemble based methods were used: the straightforward implementation and the implementation using the Sherman-Morrison-Woodbury (SMW) formula. These implementations are applicable

to the global EnKF, state based localized EnKF, global DEnKF and the state based localized DEnKF. In order to ensure that both implementations have the same effects, the implementations are subjected to an assimilation run using the same network and assimilation parameters. Both methods should provide identical model runs and identical error statistics.

First the small network that is described in Experiment 1 is used. The model runs consisted of 3600 model timesteps (of 2 seconds) and 88 cells. The model runs gave identical error statistics, as can be seen in table 6.2.

	RMSE K	MAPE K	RMSE V	MAPE V	TRE
EnKF global SF	0.0030	0.0150	0.5937	0.0050	84160
EnKF global SMW	0.0030	0.0150	0.5937	0.0050	84160
EnKF local state SF	0.0082	0.0513	1.4215	0.0149	276663
EnKF local state SMW	0.0082	0.0513	1.4215	0.0149	276663
DEnKF global SF	0.0058	0.0265	0.9660	0.0078	143549
DEnKF global SMW	0.0058	0.0265	0.9660	0.0078	143549
DEnKF local state SF	0.0059	0.0271	1.0068	0.0086	150806
DEnKF local state SMW	0.0059	0.0271	1.0068	0.0086	150806

Table 6.2: Error statistics of straightforward implementation (SF) and Sherman-Morrison-Woodbury implementation (SMW) for identical model runs on a small network

Moreover, the differences in densities of the 88 cells for the 3600 timesteps were analysed using MATLAB. The cumulative absolute error (CAE) was computed by

$$CAE = \sum_{x=1}^{X=88} \sum_{t=1}^{T=3600} \left| k_{x,t}^{(SF)} - k_{x,t}^{(SMW)} \right|.$$

The mean absolute deviation (MAD) is defined as

$$MAD = \frac{1}{X}\frac{1}{T} CAE.$$

The results are given in table 6.3. The small differences are likely caused by rounding

	Cumulative absolute error	MAD
EnKF global	4.2717e-12	1.3484e-17
EnKF local state	8.4938e-12	2.6811e-17
DEnKF global	1.1363e-11	3.5868e-17
DEnKF local state	1.2103e-11	3.8204e-17

Table 6.3: Errors of cell densities of straightforward implementation (SF) and Sherman-Morrison-Woodbury implementation (SMW) for identical model runs on a small network

errors as the densities are represented by double precision floating point numbers, as these numbers have 15-17 significant decimal digits. These rounding errors occurs when the Cholesky decomposition was used, as the matrices needed to be forced as symmetric.

Secondly, the same procedure was used for the big Rotterdam network. In order to make MATLAB cope with a reasonable memory size, only $T = 300$ timesteps were analysed with the number of cells $X = 4645$. In the large network the same conclusions hold as in the small network. The errors are too small to be of significant value.

	RMSE K	MAPE K	RMSE V	MAPE V	TRE
EnKF global	0.0039	0.7622	0.3333	0.0074	0
EnKF global SMW	0.0039	0.7622	0.3333	0.0074	0
EnKF local state	0.0021	0.3574	0.2020	0.0048	0
EnKF local state SMW	0.0021	0.3574	0.2020	0.0048	0
DEnKF global	0.0038	0.6863	0.3197	0.0070	0
DEnKF global SMW	0.0038	0.6863	0.3197	0.0070	0
DEnKF local state	0.0019	0.3208	0.1901	0.0044	0
DEnKF local state SMW	0.0019	0.3208	0.1901	0.0044	0

Table 6.4: Error statistics of straightforward implementation (SF) and Sherman-Morrison-Woodbury implementation (SMW) for identical model runs on a small network

	Cumulative absolute error	MAD
EnKF global	3.8132e-10	2.7364e-16
EnKF local state	5.8292e-12	4.1831e-18
DEnKF global	9.8418e-11	7.0626e-17
DEnKF local state	6.7514e-12	4.8449e-18

Table 6.5: Errors of cell densities of straightforward implementation (SF) and Sherman-Morrison-Woodbury implementation (SMW) for identical model runs on a large network

6.5.3 Verification of error statistics

- ✓ Computing error statistics using synthetic data and comparing the results with the expectation.
- ✓ The error statistics in the Java implementation is compared with a MATLAB computation of the error statistics.
- ✓ The correction equations of all the data assimilation methods are manually checked using synthetic input.

6.5.4 Verification of whole prototype

- ✓ When the prior estimate of all parameters are set to the true values and the associated error inputs of the parameters are set to zero, the estimated state is equal to the true state and the performance indicators give an optimal result.
- ✓ The prototype is further verified using the several data assimilation algorithms on a small network. See section 7.3 for these results.

Chapter 7

Performance of traffic estimation and prediction tool

In this chapter the performance of the prototype is tested. In the first section the setup of the experiments is described. The second section derives the main performance indicators. Sections 3 to 8 give the main results of the six performed experiments.

7.1 Experimental setup

In this section the experimental setup is described. In the first subsection the main methodology of the experiments is described. The second subsection gives an overview of the experiments.

7.1.1 Recap of design choices of prototype

In the previous chapters, the developed prototype is described and the the several design choices are further elaborated on. Here a short recap is given for the the convenience of the reader.

A model-based architecture is chosen for the estimation and prediction of the traffic dynamics. In this prototype, it is chosen to adopt the LWR model as traffic flow model, which is a fairly simple model, and measurements given by double loop detectors.

As state estimation model the Ensemble Kalman Filter (EnKF) is chosen. From the theoretical analysis, three refinements to the traditional EnKF are derived:

1. The Sherman-Morrison-Woodbury reformulation of the computation of the Kalman gain. This refinement should increase the computational speed, without loss of accuracy.
2. The deterministic approach by Sakov and Oke (2008) instead of the traditional stochastic approach. In the stochastic approach, the observations used in every en-

semble member is randomly sampled in such a way the optimal posterior covariance matrix is approximated statistically. The deterministic approach approximates the posterior covariance in a analytical way. This refinement could lead to increased accuracy without harming the computational speed too much.

3. Localization of the network. By localizing the network, the spurious correlations between physically distant model elements are deleted. Moreover, the effective ensemble size is increased. A localized EnKF should give a far better accuracy in comparison with its global counterpart.

The main goal of this experiment is to investigate the performance of the EnKF and its refinements for the use in traffic state estimation and prediction on a regional scale.

7.1.2 Experiment methodology

As main methodology to assess the performance of the prototype the (identical-)twin experiment is chosen. This methodology is widely used in data assimilation research (Yilmaz, 2015). In figure 7.1 this methodology is visualized.

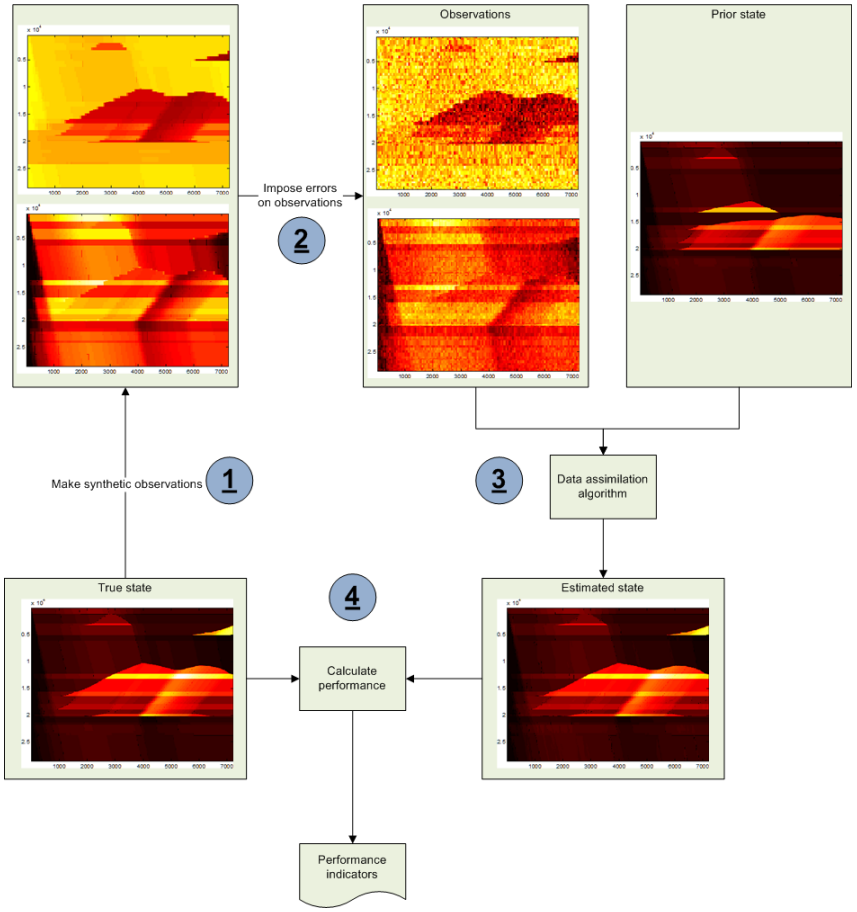


Figure 7.1: Graphical overview of the used experimental setup

The experimental setup consists of the following steps:

1. The first step is to run a simulation model to generate the "true" solution that represents real traffic flow. This simulation model generates the true traffic state, but also associated observations. In this research, the traffic state consists of the density over space and time. The associated observations are the local (1-minute smooth) speed and flow at the location of detectors. Note that the traffic state is discretized much denser than the observations: the traffic state is discretized in segments of 2 seconds and approximately 60 meter, whereas the observations are given every minute and approximately 450 meter.

2. The second step is to add some error to the true observations. This error represents the measurement noise of the detector equipment. These perturbed observations are used as the real-time observations for the data assimilation algorithm.

3. The third step is the actual assimilation. It uses the synthetic observations including the measurement noise and a prior "guess" of the traffic state. This prior traffic estimate is generated by using the simulation model with different initial conditions and parameters. It represents the imperfect knowledge one has before the assimilation. The data assimilation algorithm combines the observed data with the prior estimate of the traffic state to get the estimated traffic state.

4. The fourth step is the comparison between the estimated traffic state and the true traffic state. This is done using some performance indicators, derived in the next section.

The twin experiment framework is a suitable intermediate step towards assimilation of real observations. The main advantage of this framework is the controllability of the simulation as every step in the framework can be controlled by the modeller. One can control and investigate the influence of factors such as the measurement error, which is normally unknown. Moreover, the true state is exactly known and thus the performance of the data assimilation can be computed quite accurately. The large disadvantage of this experiment framework is the simplification of the dynamics as they would occur in real life. The model structure of the simulation model used in the data assimilation is assumed to be perfect description of the real-life dynamics. Therefore the data assimilation will perform better than it would using real-life observations.

7.1.3 Overview experiments

In figure 7.2 an overview of the experiments is given.

Experiment 1 is an experiment designed for further verification of the prototype. A small network is used in order to shorten the computation time. Therefore lots of simulations can be run in a limited time. The performance of the different assimilation methods can provide some interesting research directions in the large network. By testing some hypotheses that can be derived from literature it is tested if the prototype works as expected. Some sensitivity analyses give insight in the general working of the assimilation methods and the different trade offs.

Experiment 2 is the first experiment that uses a large network case. The first subexper-

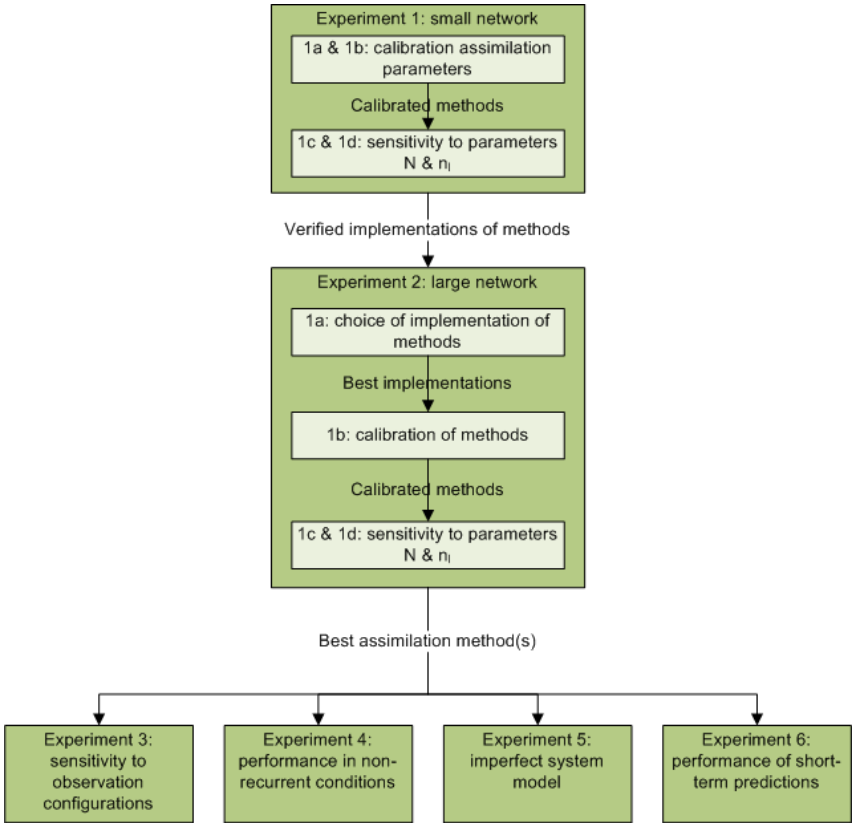


Figure 7.2: Graphical overview of the experiments

iment tries to identify the best implementation of each assimilation method. The choice of the best implementation is based on the computation times, as each implementation of an assimilation method is equivalent in terms of accuracy. In the further subexperiments the performance of the different assimilation methods are empirically compared. The goal of these subexperiments is to determine which methods perform best and are most promising for further research. This choice is based on both empirical results and theoretical reasoning. Moreover, this experiment gives a basic indication of well-performing assimilation parameters.

Experiments 3 to 6 examines the sensitivity to certain assumptions of the assimilation methods and advanced applications of the prototype. In experiment 3 the influence of different observation configurations is investigated. By observation configuration the number and location of detectors, observed variables and measurement noise are meant. This way the data assimilation methods are tested in quite extreme conditions where only a little information is provided. In experiment 4 the application of the prototype in non-recurrent conditions is tested. The performance in these non-recurrent conditions indicate the performance when unpredicted events occur and the assimilation model doesn't model or isn't capable of modeling real traffic flows. In experiment 5 this mismatch between the 'true' model and the assimilation model is further investigated. Instead of an incidental mismatch in experiment 4, in experiment 5 a structural mismatch is examined by setting an imperfect fundamental relation in the assimilation model. Experiment 6 investigates the performance in the ultimate goal of the prototype: correctly short-term predicting

the traffic.

7.2 Performance indicators

Several factors of performance are important in this experiment:

- **Accuracy of state estimation.** There exist several ways to determine the accuracy of the state estimation. One of the mainly used indicator is the root-mean-square error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{\sum (\hat{y} - y)^2}{n}},$$

where $\hat{y}$ is the predicted value, y is the true value and n is the size of the data set. The variable y can represent several quantities: e.g. density, speed or flow. Another commonly used indicator is the mean absolute percentage error (MAPE):

$$\mathrm{MAPE} = \frac{1}{n} \sum \left| \frac{y - \hat{y}}{y} \right|.$$

The MAPE gives a relative error in contrast to the RMSE that represents the absolute error. These indicators can be computed for the whole data set, but can also give more information when the data set is split into the free flow and congested parts of the data set.

Moreover, Schreiter (2013) used an indicator for analysing if congestion occurs at the same location and time:

$$J_{reg} = \sum_i \sum_j \left| \gamma^{ij} - \hat{\gamma}^{ij} \right| \Delta x^i \Delta t$$

with the traffic regime of cell i at time step j:

$$\gamma^{ij} = \begin{cases} 1 & \text{if } v^{ij} < v_c^{ij} \\ 0 & \text{else} \end{cases}$$

It is chosen to compare the density, speed and traffic regime. This leads to the following list of indicators:

1. RMSE of density k in the whole dataset
2. RMSE of speed v in the whole dataset
3. MAPE of density k in the whole dataset
4. MAPE of speed v in the whole dataset
5. Traffic regime error J_{reg} of the whole dataset
6. RMSE of density k where the traffic is free flowing

7. RMSE of speed v where the traffic is free flowing
8. MAPE of density k where the traffic is free flowing
9. MAPE of speed v where the traffic is free flowing
10. RMSE of density k where the traffic is congested
11. RMSE of speed v where the traffic is congested
12. MAPE of density k where the traffic is congested
13. MAPE of speed v where the traffic is congested

- **Accuracy of predictions using estimated state.** The accuracy of the predictions is determined by using the (mean) estimated state in order to predict the state several time steps ahead. In this case, this is done by predicting every 5 minutes the traffic state up to 1 hour ahead with 5 minute steps. The predicted states can then later be compared with the true state, using the 13 indicators described above.

- **Stability of data assimilation to initial boundary conditions.** One major requirement is that the state estimation and prediction needs to be insensitive to the error in of the boundary conditions, given a set of assimilation parameters. This means that if the network parameters are incorrect, the data assimilation will provide reasonable performance without recalibrating the assimilation parameters. In this experiment, this insensitivity can be made clear by the distribution of the performance indicators for each assimilation parameter set, when simulating with different network parameters. In particular, the worst performances are relevant. Therefore the 90th percentiles of the performance values for the different network parameters are chosen as indicator.

  Note that here the 90th percentile value is chosen as indicator, instead of the maximum value (which corresponds with the worst performance). This is due to the fact that the maximum value of the indicators isn't robust to changes in the (sample of) network parameters. That is not a significant problem in this experiment as all assimilation methods and assimilation parameters are tested with the same network parameter sets. However, choosing the maximum value as indicator would harm the comparability of the results when another sample of network parameter sets is used.

- **Computation time.** Another major factor is the computation time of the different assimilation schemes. However, for a small network the difference in computation times could be negligible. It is expected that the ensemble size is a major factor in the computation time.

7.3 Experiment 1: a small toy network using synthetic data

From the theoretical analysis some hypotheses are derived that can tested in this simulation experiment. By accuracy both accuracy in state estimation as in state prediction ($\Delta t < 1\text{h}$) is meant.

Hypothesis 1.1. The deterministic methods are more accurate than the stochastic methods.

The deterministic methods are not influenced by the sampling error of the perturbation of the observations. However, the deterministic methods use an approximation in determining the Kalman gain, which can counteract the sampling error.

Hypothesis 1.2. All methods are less accurate for smaller ensemble sizes. The deterministic methods suffer less from smaller ensembles than the stochastic methods.

A smaller ensemble will decrease the accuracy of the estimation as the linearisation becomes less accurate. Moreover, the sampling error of the stochastic methods increases as the algorithm becomes more sensitive to sampled values of the perturbed observations.

Hypothesis 1.3. The computation times of the observation based localized schemes is the lowest, followed by the state based localized schemes and the global schemes.The computation times are mostly dependent on the ensemble size.

Based on the computational complexity, this hypothesis should be true. However, it is possible that this effect can't be discovered in the small test network. One possible reason could be that the differences in computation times could be too small to be significantly detected. Another explanation could be that the computation time of the implementation doesn't match the theoretical complexity due to overhead.

The ensemble size has a large linear influence on the computation time of the prediction step of the algorithm, and the computation time of the correction step is also increased with an increased ensemble size.

Hypothesis 1.4. The smaller the radius, the less accurate the localized methods are in comparison to the the global methods.

When the localization radius becomes smaller, less information from the observations is taken into account and thereby decreasing the accuracy.

Hypothesis 1.5. There exist a configuration of an ensemble based method that has satisfying performance in both estimation accuracy, prediction accuracy, stability and computational speed.

The theoretical investigation suggest that ensemble based methods can be very helpful for the posed problem. The main goal of this simulation experiment is to investigate if an ensemble based method can perform satisfactory on a small network, and thereby also has good potential for larger networks.

In order to test these hypotheses and thereby the performance of different ensemble schemes, experiments are performed using a small test network. The small test network allows for many simulation runs as the computation time will be (relatively) short. This experiment is a so called twin experiment: firstly data is generated using the traffic model with a certain set of parameters (inflow, turning fractions, free speed etc). After perturbing the data, this data is then used as observations for runs of the traffic model with slightly different values for the parameters.

The goal of this experiment is to find a good ensemble based assimilation scheme that is accurate in a reasonable computation time. Moreover, this experiment tries to deduce some do's and don'ts for the use in the large network.

7.3.1 Experiment design

This experiment is divided into several subexperiments:

- **Experiment 1a: first calibration of assimilation schemes.** In order to fairly compare the different assimilation schemes, each scheme needs to have its parameters calibrated so all assimilation schemes perform reasonably well. This calibration is split into two parts: in experiment 1a the initial errors of the state are calibrated, and a further calibration in experiment 1b.
- **Experiment 1b: calibration of covariance inflation.** In experiment 1b, the covariance inflation factors are calibrated using the best parameters found in experiment 1a.
- **Experiment 1c: sensitivity to ensemble size.** Using the parameters found in experiment 1b, the influence of a smaller ensemble size on the performance of the assimilation is investigated. The ensemble size should have a large influence on the computation time. In order to make the trade off between accuracy and computation time, the accuracy of the data assimilation for smaller ensemble sizes is crucial.
- **Experiment 1d: sensitivity to localization width.** In experiment 1c, the assimilation schemes that perform well with smaller ensemble sizes are found. In experiment 1d, the sensitivity of these (local) assimilation schemes to smaller localization widths is investigated. For larger networks, the localization width will have significant influence on the computation time.

The design of the experiment consists of two parts: the first part is the design of the reference situation. The second part is the design of the assimilation schemes that use the observations of the reference situation in order to approximate the reference situation.

Reference situation

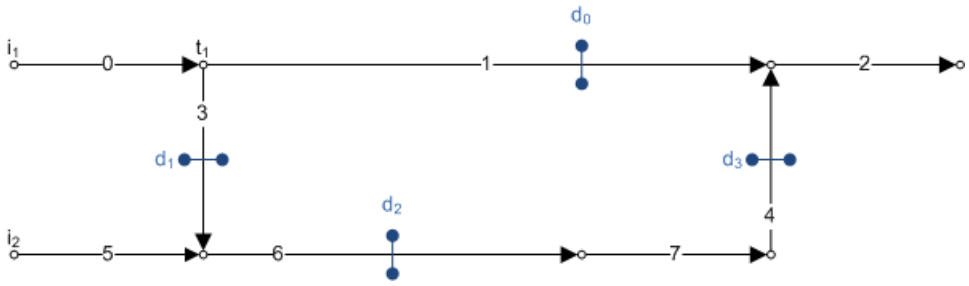


Figure 7.3: Network under consideration

In figure 7.3 the network at hand is displayed. Note that two inflow nodes and one diverge node exist corresponding inflow demands i_1 and i_2 and turning fraction t_1. The locations of the detectors are also displayed in the figure. The properties of the links are summarized in the table below.

# Link	Length	# Lanes	Speed limit
0	500 m	2	27.78 m/s
1	1500 m	2	33.33 m/s
2	500 m	2	27.78 m/s
3	500 m	2	27.78 m/s
4	500 m	2	27.78 m/s
5	500 m	1	27.78 m/s
6	1000 m	2	27.78 m/s
7	500 m	2	25 m/s

The demand patterns i_1 and i_2 are chosen as in figure 7.4. The peaks of the demand patterns are at $0.5\frac{\text{veh}}{\text{s}}$ and $0.44\frac{\text{veh}}{\text{s}}$ respectively. The turnfraction t_1 is chosen as the constant value of 0.6 in the direction of link 1. For all links, the fundamental diagram of Smulders is used with parameters $v_{cri} = 22.22$ m/s, $k_{cri} = 0.025\frac{\text{veh}}{\text{m·lane}}$ and $k_{jam} = 0.125\frac{\text{veh}}{\text{m·lane}}$ and v_{free} the speed limit of that link.

As model parameters, a time step of $dt = 2$ s is chosen. The cell lengths are chosen according to the minimum cell length as in the CFL condition: $l = dt \cdot v_{free}$. The runs are simulated for 2 hours.

In figure 7.5 the space-time plots of the reference data are given. Three routes are distinguished: route 1 consists of links $(0, 1, 2)$, route 2 consists of links $(0, 3, 6, 7, 4, 2)$ and route 3 consists of links $(5, 6, 7, 4, 2)$.

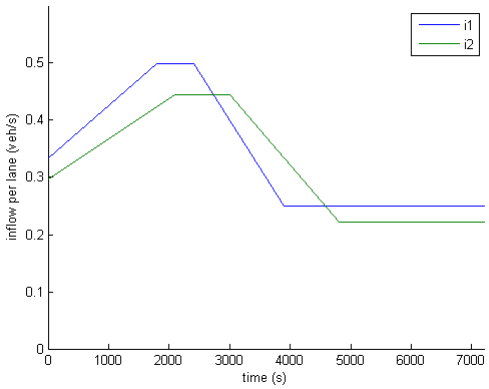


Figure 7.4: Demand patterns i_1 (blue) and i_2 (green)

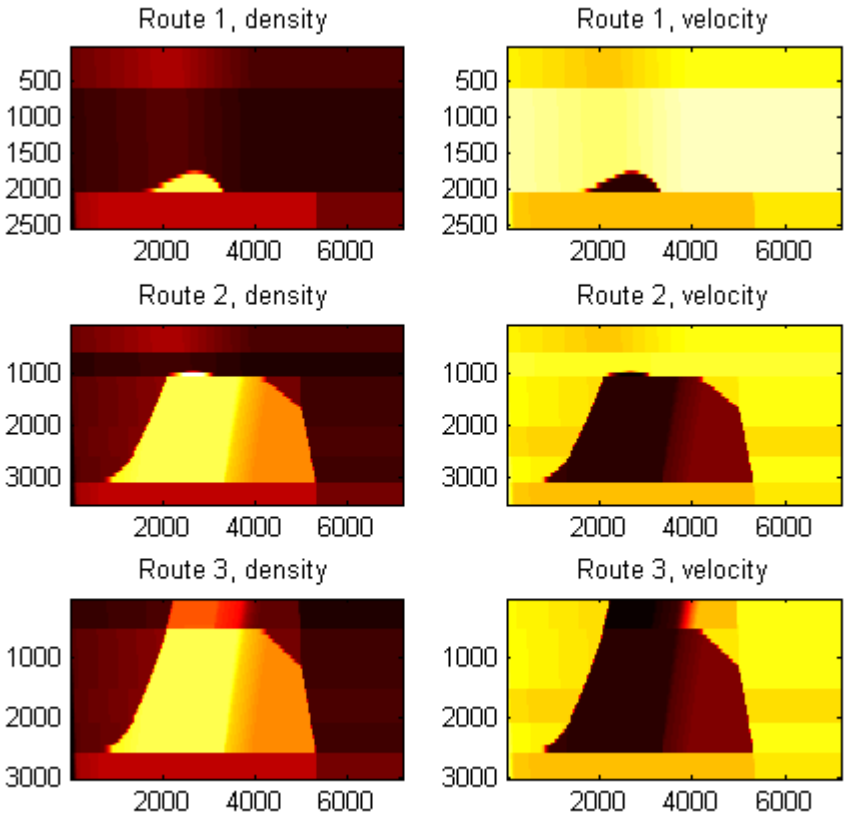


Figure 7.5: The space-time plots of the reference data.

Assimilation design

The observations are generated by storing the speeds and flows at the corresponding cells every model time step (chosen as $dt = 2$ s). Then the speeds and flows are averaged to 1-minute time mean speeds. Note that this use of the arithmetic mean is generally not correct, but corresponds to the technique used by Dutch detectors. This data is then perturbed by a Gaussian white noise with standard deviation 1.5 m/s for speed measurements and 0.04 veh/s for flow measurements in order to be used for assimilation in the experiment runs.

In this experiment, the following assimilation schemes are chosen to be compared:

1. EnKF (global)
2. EnKF (state based local analysis)
3. EnKF (observation based local analysis)
4. DEnKF (global)
5. DEnKF (state based local analysis)
6. DEnKF (observation based local analysis)
7. No assimilation (using only prior knowledge)

In Scheme 1-6 covariance inflation is excluded by setting the inflation factor to 1. The ensemble size is chosen as 20, and for the local schemes a radius of 20 cells upstream and 20 cells downstream is chosen. As the length of the cells is minimal in the sense of the CFL condition, this means that the 40 cells correspond to 1.33 minute in free speed. This is more than the update time span of 1 minute, so the localization radius should be large enough.

The assimilation schemes will be tested for 25 different demand patterns and turn fractions. The demand patterns i_1^* and i_2^* of the simulation runs are randomly chosen as $i_1^* \sim i_1 \cdot N(1, \frac{0.05556}{i_1})$ and $i_2^* \sim i_2 \cdot N(1, \frac{0.04167}{i_2})$. This means that the demand patterns for the simulation runs are scaled, where the peaks of the demand patterns have standard deviation 0.05556 and 0.04167 respectively. The turn fraction t_1^* is randomly chosen with $t_1^* \sim N(t_1, 0.15)$. All assimilation schemes are subject to the same parameters and (perturbed) observations, in order to better compare the assimilation schemes. The used parameter sets are displayed in table 7.1.

An illustrated overview of the different simulation runs is presented in figure 7.6. In this experiment 6 different assimilation schemes (excluding the 'no assimilation'-scheme) with (e.g.) 10 different assimilation parameter sets and 25 different network parameter sets are used. This leads to the execution of $((6 \cdot 10) + 1) \cdot 25 = 1525$ simulation runs.

Network parameter set	Inflow 1	Inflow 2	Turn Fraction
1	0.6133	0.4494	0.8088
2	0.4312	0.5052	0.5703
3	0.4711	0.4120	0.6747
4	0.4467	0.4230	0.5978
5	0.5821	0.4083	0.5631
6	0.4164	0.5295	0.3769
7	0.5318	0.4243	0.8145
8	0.5552	0.5073	0.6341
9	0.6046	0.4708	0.5541
10	0.4475	0.3855	0.7793
11	0.5502	0.4513	0.5721
12	0.5504	0.4059	0.4124
13	0.5390	0.4610	0.7135
14	0.5023	0.4129	0.3142
15	0.5056	0.4048	0.7705
16	0.4173	0.4666	0.8989
17	0.5249	0.3969	0.5106
18	0.4809	0.4579	0.7096
19	0.4467	0.4653	0.4057
20	0.5532	0.4048	0.6621
21	0.5360	0.4140	0.6448
22	0.5836	0.4568	0.7631
23	0.5241	0.4137	0.6457
24	0.5286	0.4749	0.7348
25	0.4619	0.4380	0.5522

Table 7.1: Overview of the used network parameter sets

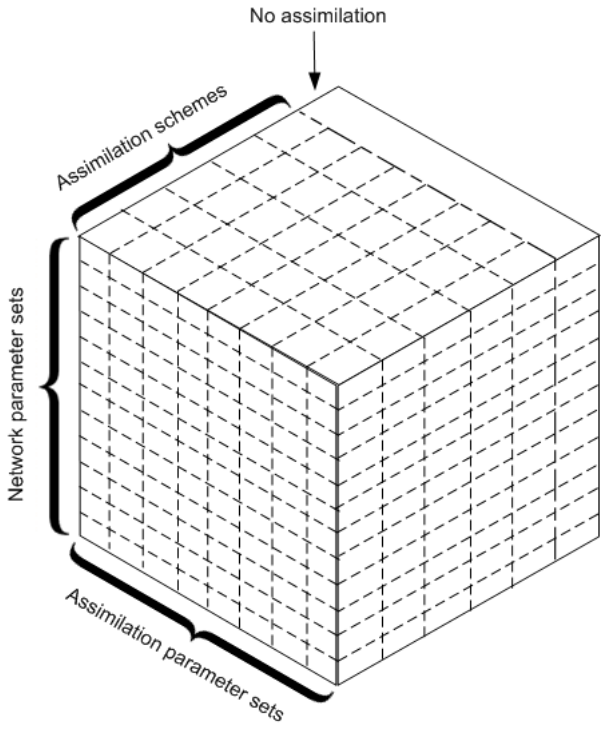


Figure 7.6: Illustrated overview of the experiment: different assimilation schemes, network parameter sets and assimilation parameter sets are compared. Note that the 'no assimilation' scheme uses only one assimilation parameter set.

7.3.2 Results experiment 1: small network

The whole overview of the results are presented in appendix B. In this subsection the main results and conclusions are given.

Hypothesis 1.1. The deterministic methods are more accurate than the stochastic methods. In table 7.2 an overview of the estimation accuracy found in experiment 1b is given, which corresponds to the best performances found in Experiment 1. The values represent the average performance over the 25 different network settings. The selected assimilation parameters are the parameters that performed best.

	RMSE K	MAPE K	RMSE V	MAPE V	TRE
EnKF global	0.0045	0.0212	0.8952	0.0104	159424
EnKF local state	0.0048	0.0229	0.9523	0.0113	178441
EnKF local observation	0.0055	0.0300	1.1020	0.0133	224706
DEnKF global	0.0045	0.0201	0.8901	0.0101	154162
DEnKF local state	0.0044	0.0202	0.8718	0.0096	148865
DEnKF local observation	0.0050	0.0235	0.9866	0.0126	177815
No Assimilation	0.0403	0.8721	7.6055	0.3921	6288372

Table 7.2: Overview mean accuracy of the assimilation methods

The absolute differences between the accuracy indicators are small. However the relative differences are quite large. Using the best assimilation parameter sets found, the global EnKF performs on average 3% worse than the global DEnKF on the main estimation perfomance indicators. The state based localized EnKF performs on average 14% worse and the observation based scores 16% worse than their deterministic counterparts. Compared to the case where no assimilation was used, the data assimilation provide a far better performance.

The simulation results thus correspond with the hypothesis: in this experiment the deterministic methods are more accurate than the stochastic methods. This can be caused by two characteristics of the DEnKF: the absence of the sampling error of the observation perturbations and the implied use of adaptive instead of a fixed covariance inflation. The increased difference between the stochastic and the deterministic methods when localized can possibly be caused by the fact that less observations are used in a localized method. The adverse effect of the sampling noise is increased when less observations are used. (A. Y. Sun, Morris, & Mohanty, 2009)

In order to visualize the performance of the data assimilation, time-space plots of an assimilation run is displayed in figure 7.7. The associated errors of figure 7.7 are displayed in table 7.3. Note that the performance for this case in quite comparable to the average performances given in table 7.2.

	RMSE K	MAPE K	RMSE V	MAPE V	TRE
EnKF global	0.0052	0.0237	0.9427	0.0079	149838
No Assimilation	0.0583	2.0211	10.6156	0.2718	11069269

Table 7.3: Error statistics of network parameter set 1, which correspond to the timespace plots in figure 7.7.

The same pattern of error statistics is prevalent in the prediction accuracy: the error of the global EnKF is 8% higher, the state localized EnKF 22% higher and the observation based EnKF scores 6% higher than their deterministic counterparts.

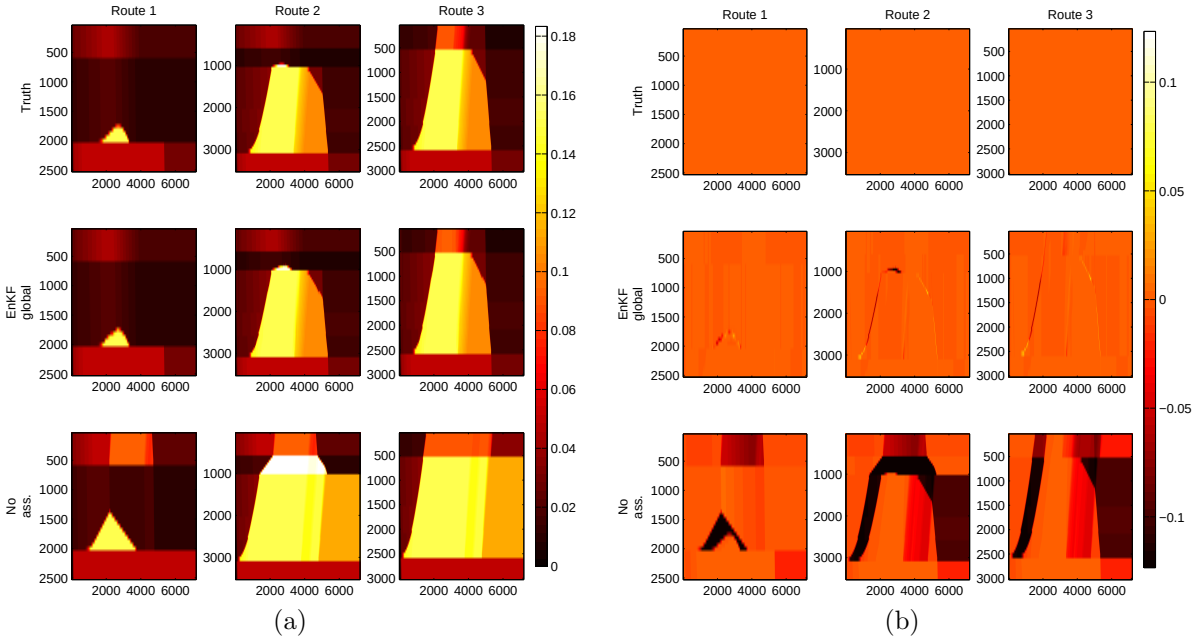


Figure 7.7: Time-space diagrams of the three routes using network parameter set 1. Depicted are the truth data, the assimilated data using the global EnKF and the no assimilation data. The horizontal axes depict time in seconds and the vertical axes space in meters. Figure (a) describes the density of the cells (in $\frac{\text{veh}}{\text{m}}$); Figure (b) describes the difference in density between the truth situation and the assimilated situations.

Hypothesis 1.2. All methods are less accurate for smaller ensemble sizes. The deterministic methods suffer less from smaller ensembles than the stochastic methods. In experiment 1c it was found that all methods become less accurate for smaller ensemble sizes. However, the influence of a smaller ensemble on the performance differs greatly. The estimation accuracies of the global DEnKF and the state localized DEnKF only decrease by circa 18% when the ensemble size is decreased from 40 to 5. The estimation accuracies of the other methods decrease by 302-723%.

The results make sense: the induced sampling noise of a stochastic method is larger for a small ensemble than for a large ensemble. This sampling noise only affects the stochastic methods. Moreover, the linear approximation made in the observation based approaches performs worse for smaller ensembles. Theoretically, (state) localization should increase the performance in small ensemble situations as localization increases the rank of the assimilation system. However, this effect is not visible in the results of this experiment.

The same result patterns hold for the estimation stability, prediction accuracy and stability.

Hypothesis 1.3. The computation times of the observation based localized schemes is the lowest, followed by the state based localized schemes and the global schemes. The computation times are mostly dependent on the ensemble size. The ensemble size ($p = 0.0000$), the assimilation method ($p = 0.0351$) and their interaction ($p = 0.0000$) were significant factors in the computation time as determined

by a two-way Analysis of Variance (ANOVA).The average computation times and linear fits (with goodness-of-fit $R^2 = 0.875$) for different assimilation schemes and ensemble size are displayed in figure 7.8.

The DEnKF approaches took more time than the EnKF approaches. Instead of what was expected, the observation based localized schemes had the largest computation time. An explanation for this could be overhead in the execution of the algorithms. The theoretical computational complexity didn't take the handling of selecting the corresponding state-observation pairs into account. The localized schemes could thus have a better theoretical performance, but in implementation the computational speed is (somewhat) slower.

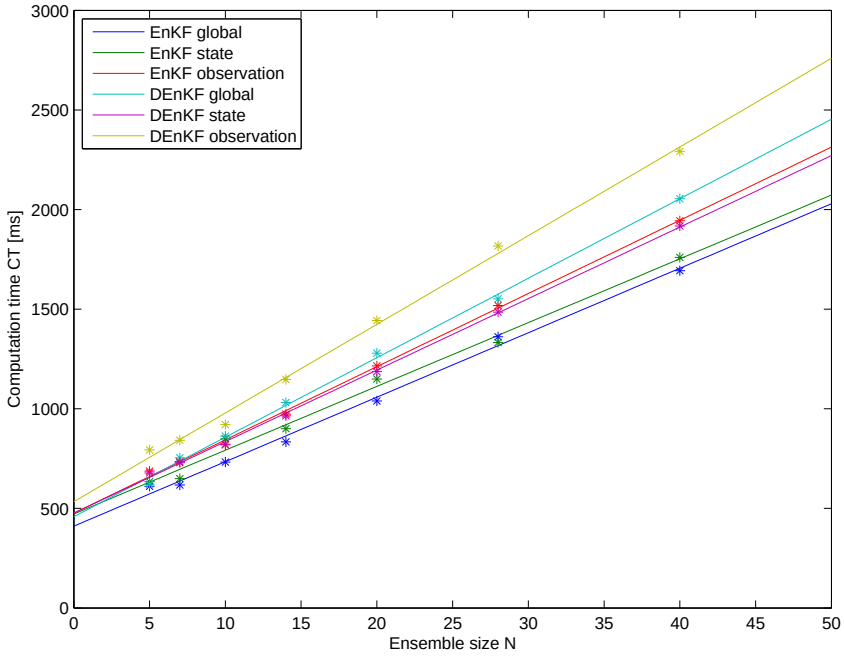


Figure 7.8: The computation times of each assimilation method for different ensemble sizes (dots) and the associated linear fits (lines)

Hypothesis 1.4. The smaller the radius, the less accurate the localized methods are in comparison to the the global methods. For this hypothesis, only the DEnKF approaches are considered as the other experiments indicated that the EnKF would perform worse for smaller ensemble sizes. Two localization widths are investigated separately: the localization width of the cells and the localization width of the inflows and turn fractions.

The results indicate that setting the localization width of the inflows and turn fractions is absolutely crucial in this network. When one of the four observations doesn't "observe" some inflows or turn fractions, the performance worsens significantly: up to 30% for only a few omitted relations. When almost no inflows or turn fractions are observed, the error indicators increase 2-fold or even 6-fold.

The importance of the correct setting of the inflows and turn fractions is also likely caused by the small size of the network. As for example route 1 consists of only 2.5 km

of road. With a free speed of 120 km/h, this means that a car travels 80% of route 1 in one assimilation time step of 1 minute. A disturbance in traffic density gets out of the network quite quickly. Therefore the dynamics of the traffic on the network at time $t + 60$s is mainly caused by the turn fractions and inflow patterns, instead of the traffic situation at time t.

The state based localized DEnKF is quite insensitive to changes in the localization width of the cells. The estimation accuracy only decreases by a few percent when the cell radius is decreased by 30%. However, the observation based localized DEnKF is more sensitive to changes in the cell radius. This could be caused by overcorrection as cells are updated multiple times when the cells lie within the influence regions of multiple detectors.

Hypothesis 1.5. There exist a configuration of an ensemble based method that has satisfying performance in both estimation accuracy, prediction accuracy, stability and computational speed. All tested assimilation methods have configurations that perform reasonably well, as can be seen from table 7.2 and the space-time plots of figure 7.7.

7.3.3 Conclusions and consequences for large scale application

The small network experiments provides good outsight for large scale application, as estimation and prediction accuracy is quite good. The theoretical benefits of localization on the accuracy of the estimation are not visible in this experiment, as the network is too small. The number of observations and number of cells is relatively small in comparison to the ensemble size.

The addition of covariance inflation works quite well, it increased the accuracy slightly in comparison to when no covariance inflation was used.

The DEnKF performs better for smaller ensemble size in this experiment. However, if this is caused by small absolute size or small relative size is unknown.

7.4 Experiment 2: Rotterdam highway network using synthetic data

In this experiment, the network is upscaled to a large network that is comparable to the network size used in practice.

In experiment 1 a simulation study was done using a small scale network. As a small network is fundamentally different in terms of importance of boundary conditions, the large model needs to be calibrated again. The benefits of localization were also not prevalent in the small network.

7.4.1 Goal and hypotheses of experiment 2

The main goal of this experiment is to investigate if the proposed framework can achieve a reasonable result on this network scale in a limited computation time. The hypotheses are split into two parts: one part covers the computational speed, and the other part handles the accuracy of the data assimilation.

Computational speed

Hypothesis 2.1. In a large traffic network, the SMW formulations provides significant benefits over the traditional formulations in terms of computational speed without loss of accuracy.

For the large scale network, several different implementations of the assimilation are possible. The Kalman gain equation can be reformulated using the Sherman-Morrison-Woodbury formula instead of the straightforward implementation. Theoretically, this application of the SMW formula will have a large impact on the computation time of the global methods and a somewhat smaller impact on the state localized methods. However, the question remains if this benefit occur in the implementation due to possible overhead.

Hypothesis 2.2. In a large traffic network, parallelization of the state localized methods provides significant benefits over single-threaded formulations in terms of computational speed without loss of accuracy.

Another option is parallelize some of the computations. By means of parallelization, independent parts of the algorithms can be computed by different cores of the (multi-core) CPU. For the global methods, the update equation ($\Delta X = K\Delta Y$) can be parallelized by the update of each ensemble member separately if K is computed. However, the benefits of parallelization would be small as this update isn't very computationally expensive: the computation of K is the computationally expensive part of the algorithm. For the state localized methods, the update of each state element is independent of the updates of the other state elements. This would thus be a natural application of parallelization. The observation based localized methods doesn't have a natural application of parallelization, as the update using one observation influences the update of the other observations.

Hypothesis 2.3. In a large traffic network, the computation of localized ensemble based methods is faster than the computation of global methods.

Theoretically, the localized methods should be faster than the global methods. However, if this still is the case in implementation remains unknown. The global methods could have a very fast computation time due to the SMW formulation. The localized methods don't have the same benefit of this formulation. Moreover, the local methods have a high chance of additional overhead due to selecting the observations and state elements from the global matrices. The state localized method could have benefits of parallelization.

Accuracy

Hypothesis 2.4. In a large traffic network, the DEnKF methods perform better than the EnKF methods.

The deterministic methods are not influenced by the sampling error of the perturbation of the observations. Therefore, the state covariance error is analytically approximated by the DEnKF, instead of statistically sampled as by the traditional EnKF. However, the deterministic methods use an approximation in determining the Kalman gain, which can counteract the sampling error. The results of experiment 1 on the small scale network suggested that the DEnKF was more accurate than the EnKF.

Hypothesis 2.5. In a large traffic network, the localized ensemble based methods are more accurate than the global methods.

In the theoretical analysis, it was identified that the localized methods benefit in two ways over the global methods. The first way is that the spurious ('fake') correlations, that are imposed due to the estimation of the state covariance by means of the ensemble, between two elements of the state or observations that are physically distant are removed. This will eliminate wrongly updating the state. The second way is that localization will increase the effective ensemble size: by splitting the ensemble into several parts one can find more combinations of ensemble members to fit the observations.

Overall

Hypothesis 2.6. A reasonable accuracy can be achieved in a limited computation time.

It is expected that a reasonable accuracy can be achieved using an ensemble based method. The local methods are the most probable methods to achieve this goal, as the theoretical benefits of these methods over the global methods in terms of accuracy are significant.

7.4.2 Experiment design

Reference situation

As in the first experiment, a reference situation is designed. The geographical description of the network including the detectors is based on the real highway network of Rotterdam. However, the traffic flows (i.e. the demand patterns, turn fractions and fundamental diagram parameters) are fictive.

The network description is based on the Dutch Regional Model (NRM). From the NRM the speed limit, number of lanes and the geographical location of the roads was extracted. The network description was then altered to better fit the use in this research. The main changes were:

- The dedicated truck lane on the A16 Northbound was merged with the lanes for other traffic, as the used traffic model doesn't consider truck traffic explicitly. Keeping the truck lane in the model would need a estimation of the portion of truck traffic for the turn fraction at that point.

- Some changes were made in the location of the (subsequent) merges and diverges in order to maintain a suitable distance between the merges and diverges. The larger distance was needed in order to make a larger time step possible that complies with the CFL condition. In this way, the minimal model time step was increased to 2.5 seconds.

- Some onramps and off-ramps were changed to delete bottlenecks at the on-ramps or off-ramps. It isn't the goal of this research to estimate the traffic state at these specific on and off-ramps. As the traffic characteristics at these locations isn't specified or calibrated correctly, and these bottlenecks could influence the whole traffic state tremendously, the bottlenecks were removed.

The detector locations were extracted from the Nationale WegenBank (NWB) via Regiolab Delft. The detector locations were automatically coupled to the links in the model. A total of 592 detector locations are used.

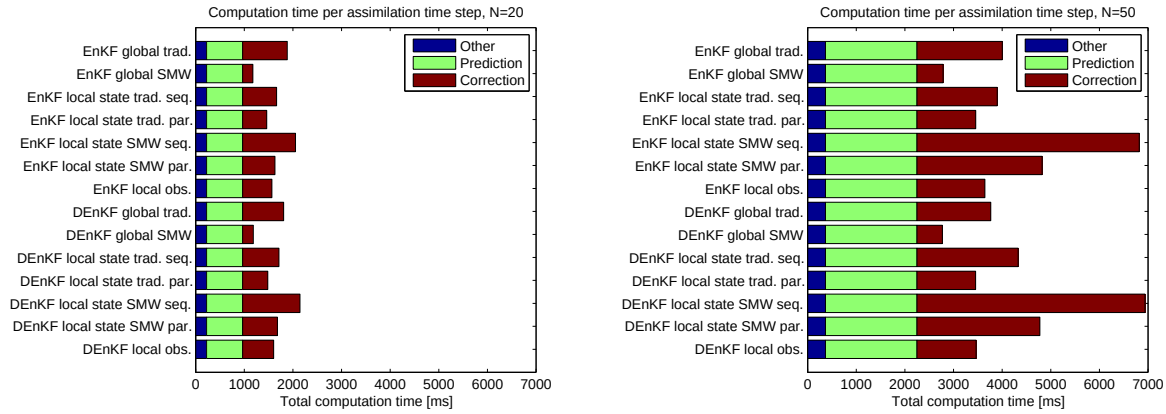
For all links, the fundamental diagram of Smulders was used with the fundamental diagram parameters were randomly chosen with small deviations around $v_{cri} = 22.22$ m/s, $k_{cri} = 0.025 \frac{\text{veh}}{\text{m·lane}}$ and $k_{jam} = 0.125 \frac{\text{veh}}{\text{m·lane}}$ and v_{free} the speed limit of that link. The demand patterns at the inflow nodes were chosen with a similar shape of the demand patterns in Figure 7.4, with the values and time instants where the shape changes randomly varied. The inflow of traffic at on-ramps was chosen as 50% of the traffic coming out of the main highway.

The turn fractions were based on the ratio of lanes per outward link. The ratio of traffic taking off-ramps was randomly chosen closely to 15%.

Assimilation design

For the assimilation models, the reference case is perturbed to serve as prior knowledge for the assimilation models. The inflows are perturbed with a standard deviation of about 16% of the values in the reference case, and the turn fractions are perturbed with 10% of the reference values.

All six assimilation methods of the previous experiment are considered in this experiment.



(a) (b)

Figure 7.9: An overview of the median computation times of an assimilation time step of the different implementations. Figure 7.9a depicts a situation where an ensemble size of $N = 20$ is used. In figure 7.9b the ensemble size is $N = 50$ is used.

7.4.3 Results experiment 2: highway network Rotterdam using synthetic data

Computational speed

In figure 7.9 the median computation times per assimilation time step for the different implementations are given. It is chosen to depict the median values, as the mean computation times suffered from severely outlying values. Both the computation times for an ensemble size of 20 (see figure 7.9a) and an ensemble size of 50 (see figure 7.9b) are given. The depicted computation times include the computation times of the prediction part of the algorithm. As hypothesized, the computation time spent is approximately linear. Furthermore, the median computation time that is needed for extra functions is displayed. This "Other" category consists of time spent at other tasks, e.g. initializing the assimilation algorithm, updating the graphical user interface, calculating performance indicators and saving data.

Hypothesis 2.1. In a large traffic network, the SMW formulations provides significant benefits over the traditional formulations in terms of computational speed without loss of accuracy. The SMW formulation provides a large increase in the computation speed for the global methods: the median computation times of the traditional implementations are almost 4 times as high as the SMW implementations for a ensemble size of 20.

For the state localized methods, the SMW formulations induced an extra overhead, which resulted in a slower computation when the SMW formulation was used instead of a traditional formulation.

Hypothesis 2.2. In a large traffic network, parallelization of the state localized methods provides significant benefits over single-threaded formulations in terms of computational speed without loss of accuracy. As can be seen in figure 7.9, parallelization provides a quite substantial increase in the computational speed. Depending on the implementation, the correction step is computed about 30% faster.

Hypothesis 2.3. In a large traffic network, the computation of localized ensemble based methods is faster than the computation of global methods. Surprisingly, the global methods are the fastest methods. This is caused by the large improvement made by the SMW implementation. Moreover, the overhead induced by the selection of the right state-observation combinations most likely plays a role in the computation time of the localized schemes.

Accuracy

	Set	RMSE K	MAPE K	RMSE V	MAPE V	TRE
EnKF global	5	0.0477	0.3264	5.1230	0.5090	114212194
EnKF local state	5	0.0085	0.0548	1.2257	0.0224	10774534
EnKF local observation	5	0.0114	0.0758	1.4757	0.0390	14360733
DEnKF global	2	0.0436	0.3236	4.6697	0.4909	101884872
DEnKF local state	5	0.0025	0.0284	0.4127	0.0085	2187272
DEnKF local observation	5	0.0030	0.0327	0.4837	0.0057	2853298
No Assimilation	-	0.0518	0.3917	5.6012	0.5141	135456313

Table 7.4: Mean performance of state estimation using different assimilation schemes.

In table 7.4 an overview is given of the mean performance of the different assimilation methods over three different starting points of the demand and turn fractions (which is input as prior knowledge). The table indicates the best of 10 different sets of (assimilation) parameters such as the initial errors and covariance inflation. The ensemble size was set as $N = 20$, and the localization radii were set to $r_c = 20, r_{i,tf} = 60$.

Increasing the ensemble size increases the accuracy, however the pattern of which method performs best doesn't change. The chosen localization radii was not optimal: the localized methods performed better when the localization radius of the inflows and turn fractions were decreased. In terms of the comparison of the assimilation methods, when the localization radii were chosen optimally the same pattern occurs as in the table above, although the differences in accuracy between the methods are very small. The state based approaches and the deterministic approaches are however less sensitive to non-optimal chosen parameters.

Hypothesis 2.4. In a large traffic network, the DEnKF methods perform better than the EnKF methods. On the basis of the performance indicators and in table 7.4 and the space-time plots in figure 7.10, the DEnKF methods perform slightly better than the EnKF methods. However, the difference in accuracy is small. As the calibration of the methods is executed quite crudely, it isn't guaranteed that the difference in accuracy isn't caused by the choice of assimilation parameters.

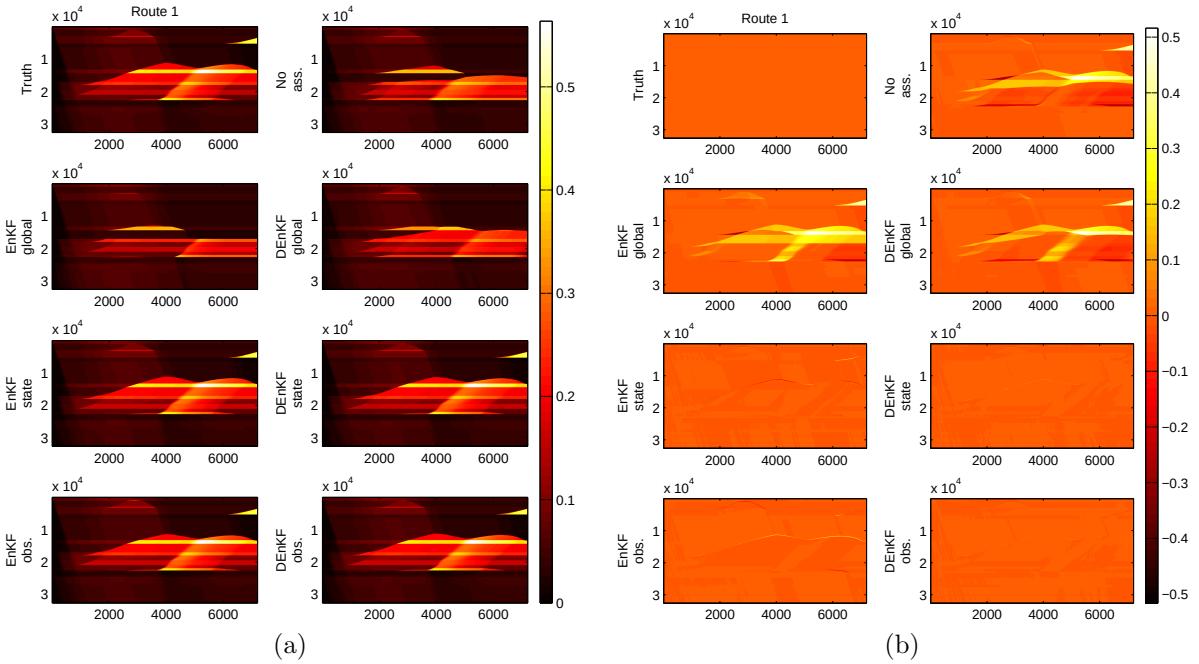


Figure 7.10: Time-space diagrams of a route using different assimilation methods. The horizontal axes depict time in seconds and the vertical axes space in meters. Figure (a) describes the density of the cells (in $\frac{\text{veh}}{\text{m}}$); Figure (b) describes the difference in density between the truth situation and the assimilated situations.

Hypothesis 2.5. In a large traffic network, the localized ensemble based methods are more accurate than the global methods. Based on the performance in this experiment, the localized methods perform far better than the global methods. The global methods are not useful for this application, as a severe mismatch between the estimated and true location of the congestion exists. This result is in line with the theory: the global methods are not suitable for a large application as this. The observation based localization performs slightly worse than the state based localization.

Overall

Hypothesis 2.6. A reasonable accuracy can be achieved in a limited computation time. In figure 7.11 the computation time is compared with the accuracy (in terms of the RMSE of the density) of the different assimilation methods.

The local approaches definitely perform better than the global approaches. Although the global methods have a faster computation time, the accuracies of the global methods are not even close to the accuracy that other methods easily reach. In comparison to the error of doing nothing, the global methods do not perform very well. The DEnKF methods consistently perform better than their stochastic counterparts.

On basis of this graph, the state based localized DEnKF performs best.

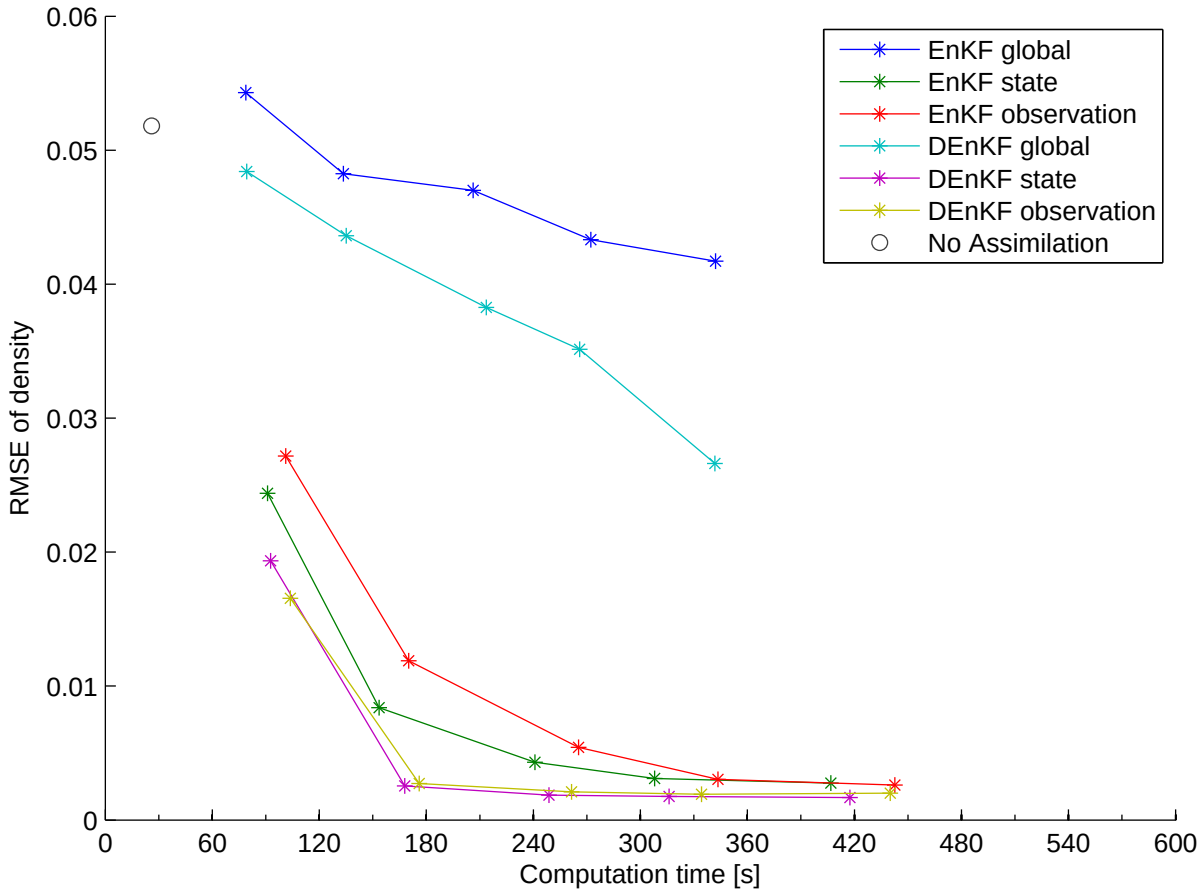


Figure 7.11: The RMSE of the density of the cells, against the computation time of 2 hours of data. The stars indicate the used ensemble size: from left to right ensemble sizes of 10, 20, 30, 40 and 50 are used

Conclusion and discussion experiment 2

The main goal of this experiment was to investigate if the proposed framework could achieve reasonable results in terms of computation time and accuracy on a large network scale. Six different ensemble based methods were tested: the traditional EnKF and the DEnKF were both used in a global, state localized and observation localized context.

In terms of computation time, it is possible to run the ensemble based algorithms in more than 10× real-time on a mid-end household computer, while using a reasonable ensemble size. The global methods have the fastest computation times due to the Sherman-Morrison-Woodbury reformulation. The localized methods were somewhat slower, as the overhead induced by the localization was of significant impact. The observation based localization is slower than the state based localization, as the state based localized algorithm can be (easily) parallelized.

In terms of accuracy, the localized methods perform far better than the global methods ceteris paribus. This is in line with the theoretical considerations, as the localization increases the effective ensemble size. The observation based localized methods perform

slightly worse than the state based localized methods. This can be caused by the additional approximation involved in the observation based scheme.

From this experiment the preferred localization scheme can be deduced. Although the computational speed of the global scheme is higher, the accuracy of the global scheme is too bad to be useful. The accuracy of observation based localization is found to be slightly lower than the state based localization on both theoretical and empirical results. Moreover, the computational speed of the observation based localized scheme is in practice lower than the state based method due to the lack of parallelization. Therefore it is concluded that the state based localized methods perform best.

The DEnKF seems to score slightly better than the traditional EnKF in the empirical results. However, the choice between the traditional method and the deterministic method can't be made yet. The theoretical benefits are unclear, as the avoided sampling error by the DEnKF is countered with an (analytical) approximation of the posterior state covariance. The improved accuracy of the DEnKF can be caused by external factors as the limited calibration. Moreover, it is not clear if the DEnKF still performs better when considering other more realistic situations, such structural mismatch between the true and assimilation system model, large errors in observations and non-recurrent events.

7.5 Experiment 3: sensitivity to observation configurations

In experiment 2, six different assimilation methods were tested on a large-scale network on basis of accuracy and computational speed. It was found that the state localized assimilation methods performed much better than the global methods and slightly better than the observation localized methods. This conclusion was drawn on basis of the accuracy - computational speed ratio: the state localized approaches were more accurate when given the same computation time.

Although in the previous experiments the DEnKF approach performed better than the traditional EnKF approach, the question remains if this is the case in different circumstances as this conclusion doesn't have a solid theoretical basis. For example, this conclusion possibly doesn't hold in situations with a limited number of observations. This experiment investigates the sensitivity of the estimation accuracy with respect to the configuration of the used observations.

7.5.1 Goal and hypotheses of experiment 3

By configurations is meant:

- Detector locations.
- Number of detectors.
- Measured variables.

- Errors in measurements.

This leads to the following hypotheses.

Hypothesis 3.1. The use of less detectors leads to less accurate estimation results.

The detectors in the Rotterdam region are very densely placed: in average a detector is placed every 450 meter. In the future, the number of detectors will possibly decrease to a decrease in available budget. It is preferable if an assimilation method isn't sensitive to the number of detectors. It is expected that a smaller number of detectors will lead to less accurate results, as the amount of (non-conflicting) information fed into the assimilation algorithm is decreased.

Hypothesis 3.2. The spread of detectors over different links leads to more accurate estimation results compared to more detectors on the same link.

It can be beneficial to place the detectors correctly, e.g. assuring detectors are present between on-ramps and off-ramps. This decreases the degrees of freedom of the state to the observations, as the detector values are less dependent on each other.

Hypothesis 3.3. Observing both velocity and flow benefits the estimation accuracy over observing only one of these variables.

By measuring both velocity and flow, one can better identify the right state. For example a flow value corresponds to two density values: one in the free flow branch and one in the congested branch of the fundamental diagram. By using the speed observations, this choice can be made.

Hypothesis 3.4. Large random errors in the observation values leads to less accurate estimation results.

Large errors in the observation values corresponds to less certainty in the observed values. This means that the additional information added by the observation to the state is relatively small.

Hypothesis 3.5. The state based DEnKF performs better than the state based traditional EnKF in all these observation configurations.

In experiment 2, the state based DEnKF performed slightly better than the state based traditional EnKF. The question remains if this also holds in this experiment. One can expect that due to the more extreme configurations (less, reliable and worse spread observations) the difference between the prior state and the posterior state becomes larger. As the DEnKF is based on the approximation that this difference is small, the question remains if the DEnKF is still preferable in this situation.

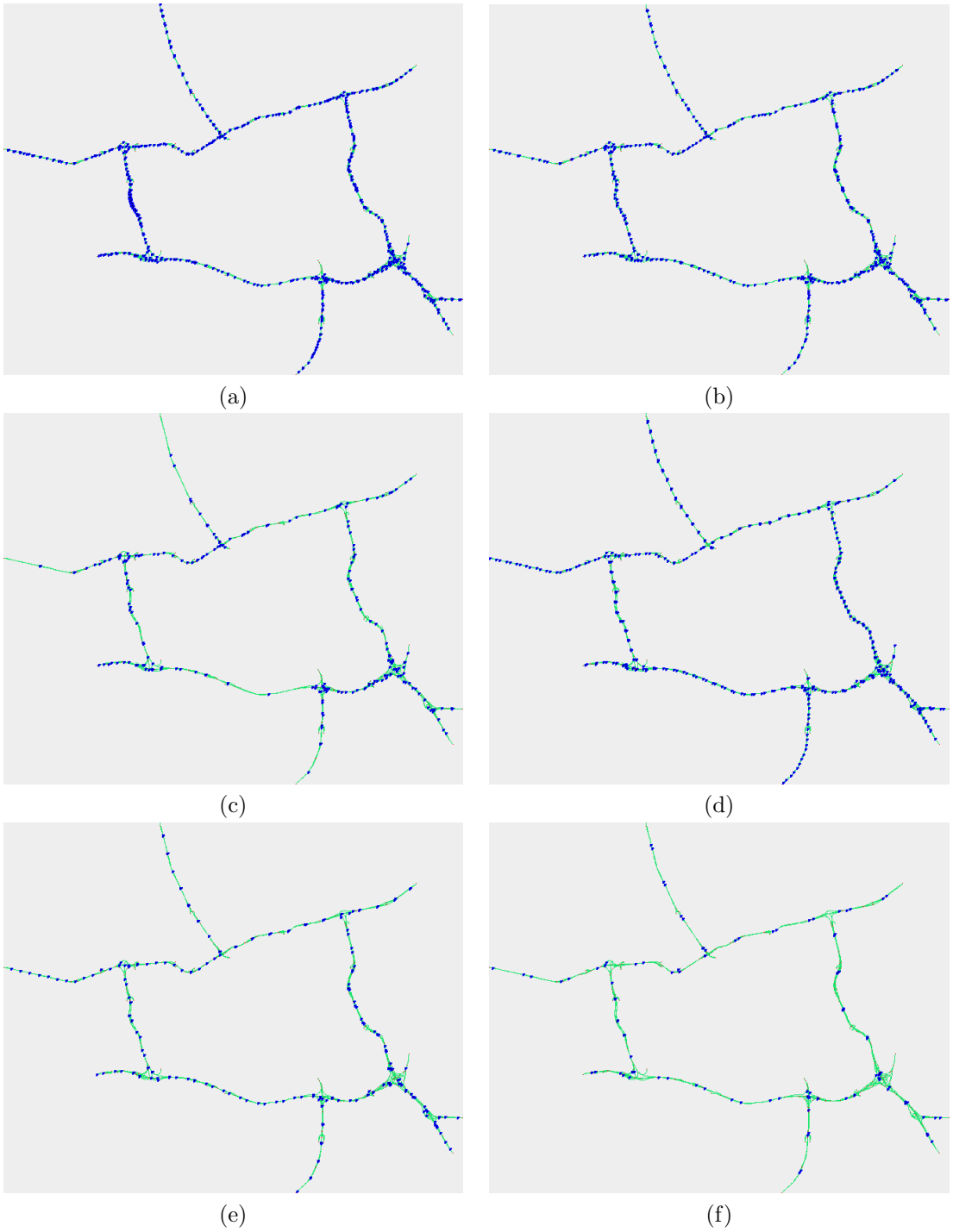


Figure 7.12: The six different detector location configurations:
Figure (a) is configuration 1 (real detector locations).
Figure (b) is configuration 2 (intermediate subset of real detector locations)
Figure (c) is configuration 3 (minimal coverage of links by real detectors)
Figure (d) is configuration 4 (same number of detectors as (b), evenly spaced)
Figure (e) is configuration 5 (same number of detectors as (c), evenly spaced)
Figure (f) is configuration 6 (very small number of evenly spaced detectors)

7.5.2 Experiment design

Detector locations

In table 7.5 the six different detector location configurations are shown. Two types of configurations are used: the first group consists of the configurations that use the real detector locations, whereas the configurations in the second group use (fictional) detector locations that are evenly spaced.

Configuration 1 is the configuration where the detectors are located as in the real Rotterdam network. This detector network is quite dense, as 592 detectors are placed within a 264 km network. This configuration is used as the base case for the other configurations. Configuration 3 is the "minimal coverage of links" configuration: for every link which has 1 or more detectors in the first configuration, only 1 detector is chosen for that link. The term link is here defined as an uninterrupted (by on- or off-ramps) road stretch that has similar characteristics (e.g. number of lanes or speed limit). Configuration 3 thus always has a detector on a link if that link has a detector in the real network. Configuration 2 is an intermediate configuration that lies in the middle between configuration 1 and 3 in terms of the number of detectors.

The second group of detector location configurations consists of configurations with evenly spaced detectors. These configurations are generated by placing detectors after a certain spacing interval. In order to ensure a fair comparison with the other configurations, only links that have detectors in the real network are considered. This way it is prevented that a lot of detectors are placed on on- and off-ramps that are not available in the real network. Note that, in contrast to configurations 2 and 3, some links that have detectors originally don't have detectors in these detector location configurations. This is a crucial difference, as the estimation of inflows and turn fractions will become a lot harder as the influence of these variables can't be separated by the filter. Configuration 4 and 5 are generated for comparison with configuration 2 and 3 as they have the same number of detectors. Configuration 6 is used as a extreme case, with detectors spaced 2000 meters.

Group	Configuration	Configuration ID	Detectors [#]
Real detector locations	All detectors	1	592
	Intermediate number of detectors	2	410
	Minimal coverage of links	3	227
Evenly spaced locations	Intermediate number of detectors	4	410
	Small number of detectors	5	227
	2000 m spaced detectors	6	111

Table 7.5: Overview characteristics used detector location configurations

Measured variables and random errors

The detectors have three measurement options: they can observe the flow, velocity or both. In this experiment/prototype, the omission of e.g. the velocity measurements is modeled by setting the associated observation errors in the matrix R to a very large

number ($\approx 1.80 \cdot 10^{308}$, which corresponds to the maximum value of a double precision floating point number). The reason for this is the ease of implementation. However, a disadvantage is the higher computation time in comparison to just omitting the measurement. This way, the computation time of selecting only one observation variable can't be measured.

Two settings of the random errors are used: the setting used in the previous experiments of 2.25 m^2/s^2 and 0.0016 veh^2/s^2, and an increased random error of 22.5 m^2/s^2 and 0.016 veh^2/s^2.

Experiment configuration

The same reference case is used as in the previous experiments. The state based EnKF and state based DEnKF are tested, using calibrated parameters found in the previous experiment..

7.5.3 Results of experiment 3

The results of all cases can be found in appendix D.

	Q & V				Q				V			
	Real locations		Evenly spaced		Real locations		Evenly spaced		Real locations		Evenly spaced	
# Detectors	EnKF	DEnKF	EnKF	DEnKF	EnKF	DEnKF	EnKF	DEnKF	EnKF	DEnKF	EnKF	DEnKF
592	0.0023	0.0021	-	-	0.0034	0.0027	-	-	0.0082	0.0078	-	-
410	0.0029	0.0025	0.0038	0.0026	0.0053	0.0028	0.0108	0.0046	0.0105	0.0096	0.0094	0.0102
227	0.0084	0.0089	0.0072	0.0054	0.0098	0.0100	0.0135	0.0082	0.0146	0.0152	0.0156	0.0162
111	-	-	0.0172	0.0197	-	-	0.0266	0.0293	-	-	0.0223	0.0206

Table 7.6: Overview performances using a small observation error

	Q & V				Q				V			
	Real locations		Evenly spaced		Real locations		Evenly spaced		Real locations		Evenly spaced	
# Detectors	EnKF	DEnKF	EnKF	DEnKF	EnKF	DEnKF	EnKF	DEnKF	EnKF	DEnKF	EnKF	DEnKF
592	0.0047	0.0051	-	-	0.0075	0.0067	-	-	0.0133	0.0126	-	-
410	0.0069	0.0065	0.0067	0.0070	0.0103	0.0096	0.0115	0.0110	0.0143	0.0143	0.0137	0.0140
227	0.0122	0.0107	0.0125	0.0111	0.0246	0.0227	0.0188	0.0197	0.0187	0.0188	0.0206	0.0198
111	-	-	0.0200	0.0177	-	-	0.0299	0.0320	-	-	0.0263	0.0243

Table 7.7: Overview performances using a large observation error

Hypothesis 3.1. The use of less detectors leads to less accurate estimation results. The results suggest that better performances are reached when more detectors are used.

Hypothesis 3.2. The spread of detectors over different links leads to more accurate estimation results compared to more detectors on the same link. The impact of the spacing of the detectors is not clear. The evenly spaced detectors were expected to perform worse than the detectors placed at real locations, as more variation in the observed values were expected when using the real locations. This conclusion seems to hold when lots of detectors are available. However, when relatively few detectors are selected, the results are not clear: the performance of the evenly spaced detectors was for some configurations worse, but for other configuration better.

Hypothesis 3.3. Observing both velocity and flow benefits the estimation accuracy over observing only one of these variables. Observing both velocity and flow leads to better results than observing only one of these variables. In terms of the comparison of observing only the velocity and only observing the flow, this experiment gives no good answer. This experiment suggests that it is better to use only the flow observations than only the velocity observations, except when very few measurements are available. This is probably caused by the setting of the observations errors: the flow observations were assumed to be relatively much more reliable than the velocity observations.

However, also theoretical arguments can be given for the found pattern: due to the shape of the fundamental diagram of the flow, a relatively large uncertainty in flow corresponds to only a small uncertainty in density. This is contrary to the velocity situation, where a large uncertainty in velocity corresponds to a large uncertainty in density. For the estimation of the density, the flow observations are thus more suitable.

For situations where only a few observations are available, little confidence is put in the state and the ensemble is widely spread. In this case, the speed observations may be better as the speed-density relation is "more linear" than the flow-density relation. Therefore, the update of the state will be more accurate. Moreover, the linearisation through widely spread flow observations can become nearly horizontal, which means that the flow error corresponds to a very large density error. In that case the flow observations add no value to the estimation of the density.

Hypothesis 3.4. Large random errors in the observation values leads to less accurate estimation results. As hypothesized, larger errors in the observations correspond to worse estimation performance. The main patterns in performance of the different configurations stay the same for the higher observation errors.

Hypothesis 3.5. The state based DEnKF performs better than the state based traditional EnKF in all these observation configurations. In most cases, this hypothesis seems to hold, as the DEnKF has a lower value of the RMSE of the density. However, in some cases, e.g. when a small number of observations is used, the DEnKF has a higher error than the EnKF. This situation is further analysed.

Although the RMSE of the density is higher, the general shape of the space-time plots in figure 7.13 seem to indicate that the DEnKF assimilation seems to perform far better than

the EnKF method. The congestion pattern of the DEnKF better approximates the true congestion pattern, where as the congestion pattern of the EnKF has more resemblance with the no-assimilation congestion pattern.

The DEnKF suffers from quite severe oscillations in estimated density. This causes the performance to be quite low. These oscillations can be caused by several effects:

1. Due to the lack of observations, the ensemble is widely spread and the inaccuracies produced by the non-linearity of the process model as described in subsection 5.2.3 are high and will produce "artificial updates". Moreover, the wide ensemble spread will cause the linearisation of the fundamental diagram to be inaccurate. This effect was mainly found in the first update steps.
2. The state is overcorrected every update due to the lack of confidence in the model state. This effect was found in the situation at hand: the speeds and flows around an observation were close to each other, in contrast to the inflow of an on-ramp nearby. A difference between the predicted observation and the observed value thus leads to a large update of the inflow, which gave a large effect on the traffic state in following time steps.
3. Possibly the approximation on which the DEnKF is based fails as the difference between the prior state and the true state is too large.

This issue could be possibly fixed by:

1. Better calibration. For example changing the initial error values of the state variables. This way the ensemble spread is small and hopefully stay small.
2. Changing the localization. As the number of observations decreases, the localization scheme can have some negative influence on the accuracy. By localizing the assimilation scheme with the use of a few observations, the update of a state element is governed by only one or few observations. Therefore, the update of the state is very dependent on that observation. By setting other localization parameters or smoother localization the (negative) influence of the localization will be contained.

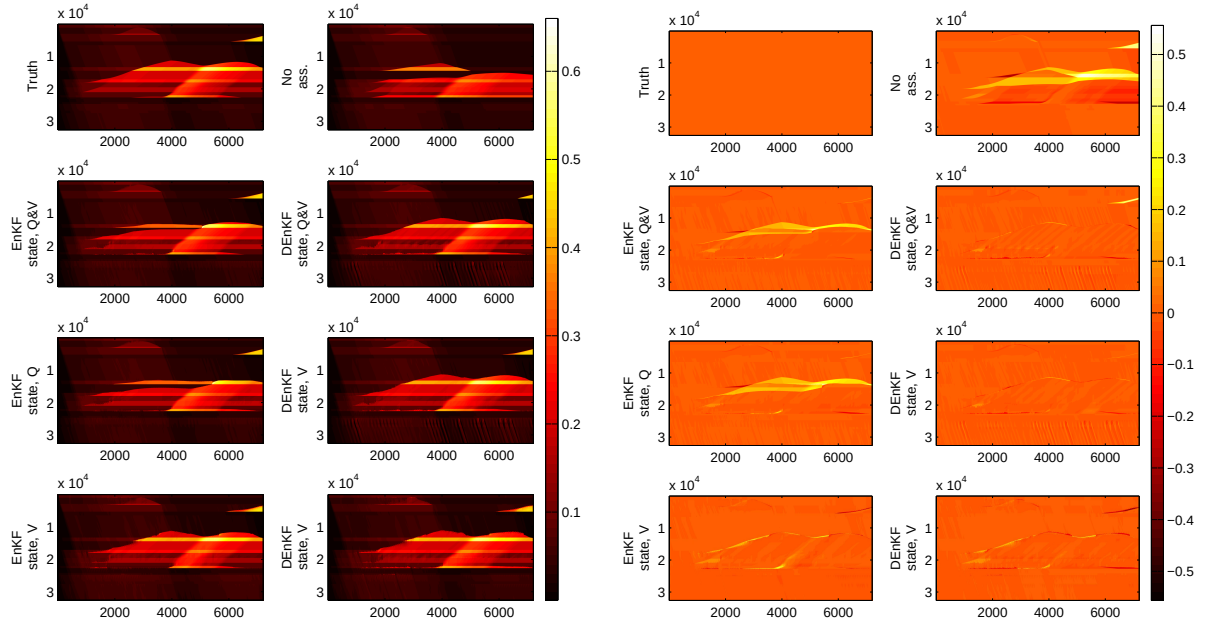


Figure 7.13: Time-space diagrams of a route using different assimilation methods. The horizontal axes depict time in seconds and the vertical axes space in meters. Figure (a) describes the density of the cells (in $\frac{\text{veh}}{\text{m}}$); Figure (b) describes the difference in density between the truth situation and the assimilated situations.

7.6 Experiment 4: performance in non-recurrent conditions

One of the main applications of the estimation and prediction tool is the application in non-recurrent conditions, as the traffic operator is most interested to use the tool in these conditions.

7.6.1 Goal and hypotheses of experiment 4

One would want to have the algorithm adaptable to non-recurrent and unpredictable conditions. Examples of these unpredictable conditions are accidents and malfunctioning of infrastructure. These conditions or events indicate a sudden end of a trend of the system model: suddenly congestion appears that could not be predicted moments before.

If the algorithm puts too much confidence in its own state and associated process model, the input of conflicting observations due to the unpredicted conditions are possibly ignored. The algorithm identifies the changed observations as measurement noise instead of changing the state.

Hypothesis 4.1. More information available to the assimilation model leads to better assimilation results

One major factor in the estimation of the state in these events is the amount of information about the event that is available and supplied to the assimilation model. One can think about information about the number of lanes and associated capacity decrease at an incident location, or even changed routing information.

Hypothesis 4.2. In non-recurrent conditions, the state based DEnKF estimates the state more accurately than the state based EnKF.

Based on the previous experiments, the state based DEnKF performs better than the state based traditional EnKF. The question remains if this is also the case in the case of non-recurrent conditions. As the approximation that is the basis of the DEnKF only holds in case of relatively small updates, the accuracy of the DEnKF could possibly be hampered by the large corrections that need to be made in the non-recurrent events.

7.6.2 Experiment design

Reference case

As an example of a unpredictable condition the malfunctioning of the Van Brienenoord bridge is used. This bridge is one of the most used bridges in the Netherlands. In 2014 the bridge malfunctioned multiple times, which caused the Van Brienenoord bridge to fail its closing procedure. (OmroepWest, 2014)

The following scenario is chosen:

At a certain point in time, the bridge opens. The procedure of opening and closing takes about 15 minutes. The north-to-south connection is then clear to travel, however the south-to-north connection is stuck and will remain closed for traffic for another 45 minutes. In the first 15 minutes during the normal closing time, only the traffic near the bridge will take the off-ramp as to divert from the open bridge. The traffic further away from the bridge will not change its behaviour, as no indication exists that something is wrong.

After the initial 15 minutes, the traffic is notified that the bridge is unavailable. Two major diversion routes exists: one through the urban roads via the Maastunnel, and one through the Beneluxtunnel on the A4. The reference case is modified in such a way that these diversion routes are more extensively used.

Assimilation case

The malfunctioning of the bridge can be incorporated in three ways:

1. No input to the assimilated models. In this scenario, the events that occur are not incorporated into the assimilated models. It is unlikely that this approach gives reasonable results, as the assimilation method can't change the state in a satisfactory manner: i.e. the capacity isn't part of the state, so it can't be changed by the assimilation algorithm.

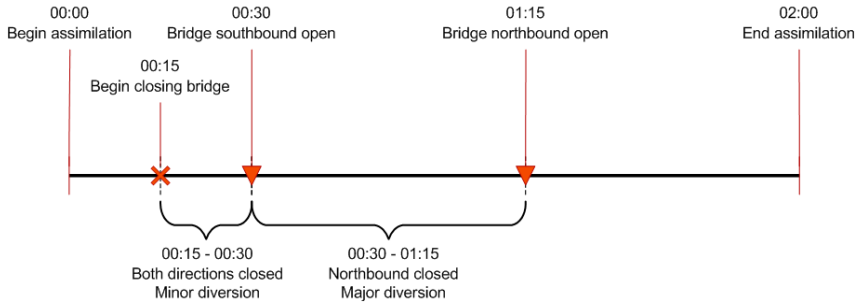


Figure 7.14: Timeline of the bridge closing scenario

2. Only the closing of the bridge is incorporated in the model. This means that the capacity on the bridge is set to zero, and the rest of the network is unchanged. The assimilation method should figure out the changes in turn fractions on its own. The time duration of the obstruction should be known.

3. The closing of the bridge is incorporated, with addition of a rough estimate of change in turn fractions. This rough estimate can be based on previous experiences with such diversion routes. The time duration of the obstruction should be known.

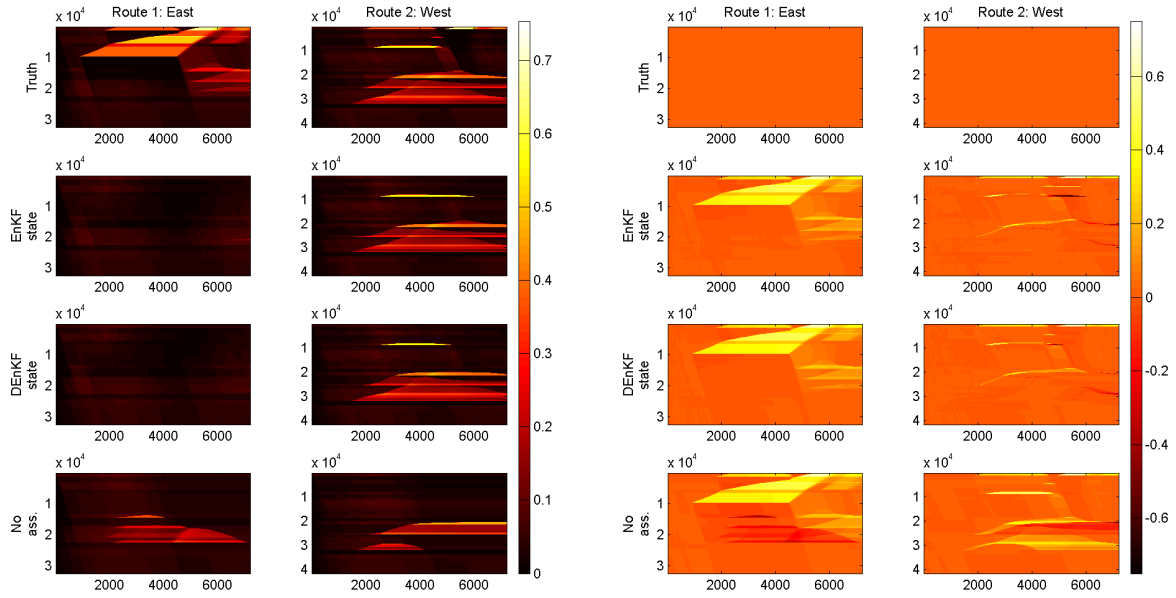
7.6.3 Results of experiment 4

Input	Method	Set	RMSE K	MAPE K	RMSE V	MAPE V	TRE
1: no information	EnKF local state	9	0.0661	0.2704	6.1951	0.4973	139614831
	DEnKF local state	9	0.0657	0.2658	6.1432	0.4712	138116751
	No Assimilation	-	0.0761	0.6048	7.4234	0.6051	226681093
2: road closing	EnKF local state	6	0.0362	0.2545	3.3011	0.1458	42582966
	DEnKF local state	9	0.0320	0.1940	3.1072	0.1217	41283309
	No Assimilation	-	0.0594	0.5898	5.5823	0.2937	132230278
3: road closing + estimate route diversions	EnKF local state	5	0.0116	0.0628	1.5086	0.0344	12896963
	DEnKF local state	9	0.0079	0.0415	1.0791	0.0217	7181606
	No Assimilation	-	0.0497	0.4232	5.0643	0.2925	113967932

Table 7.8: Overview performance of different scenarios. Each scenario adds more information to the assimilation model.

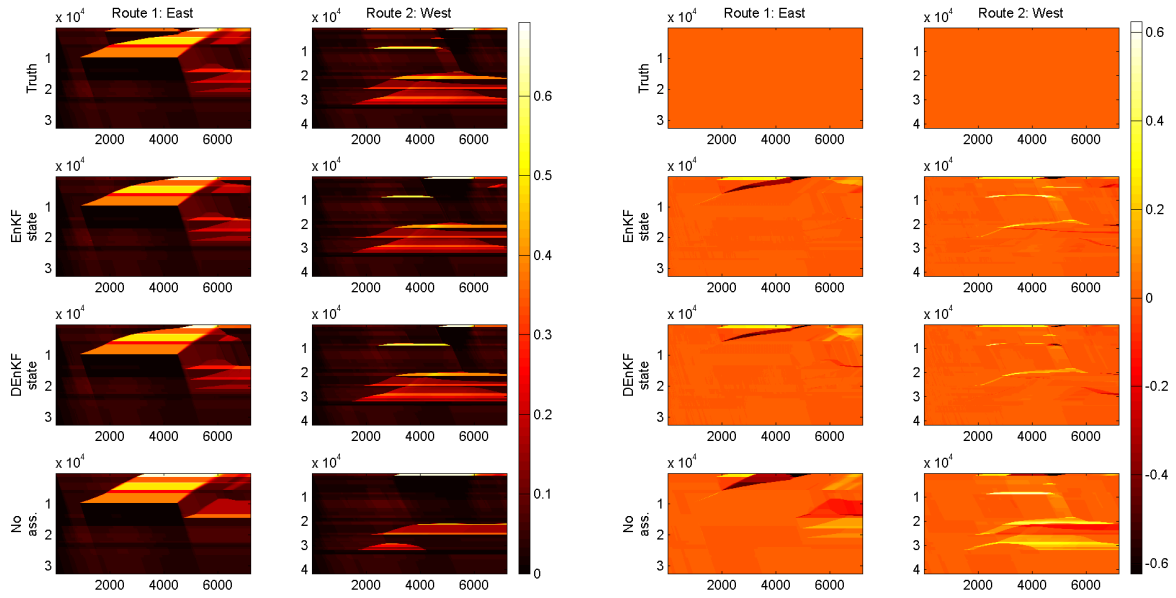
Hypothesis 4.1. More information available to the assimilation model leads to better assimilation results In table 7.8 an overview is given of the performances in the three described different scenarios. It is obvious that the addition of more (consistent) information leads to better estimation results.

The first scenario performs very bad. This was expected, as no capacity estimation procedure is used in the data assimilation algorithm. Therefore the only way to fit the zero capacity caused by the closed bridge by setting the turn fraction upstream to 100% the other way. This way the congestion upstream of the closed bridge is not estimated at all, which is of course not useful in practice.



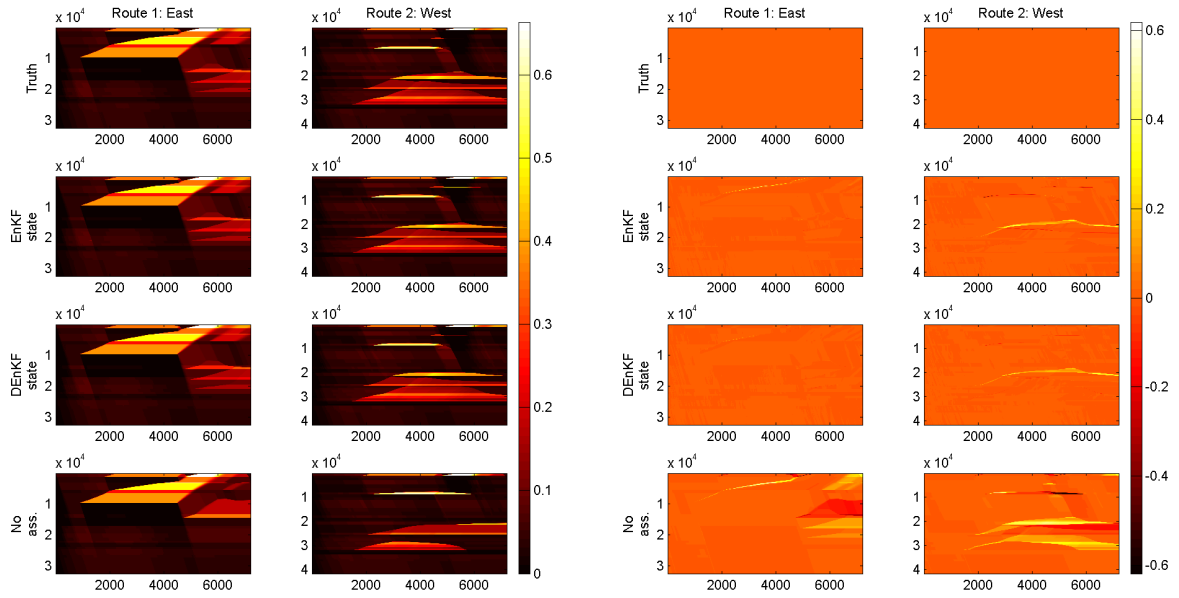
(a) (b)

Figure 7.15: Time-space diagrams of two routes using different assimilation methods. The horizontal axes depict time in seconds and the vertical axes space in meters. Figure (a) describes the density of the cells (in $\frac{\text{veh}}{\text{m}}$); Figure (b) describes the difference in density between the truth situation and the assimilated situations.



(a) (b)

Figure 7.16: Time-space diagrams of two routes using different assimilation methods. The horizontal axes depict time in seconds and the vertical axes space in meters. Figure (a) describes the density of the cells (in $\frac{\text{veh}}{\text{m}}$); Figure (b) describes the difference in density between the truth situation and the assimilated situations.



(a) (b)

Figure 7.17: Time-space diagrams of two routes using different assimilation methods. The horizontal axes depict time in seconds and the vertical axes space in meters. Figure (a) describes the density of the cells (in $\frac{\text{veh}}{\text{m}}$); Figure (b) describes the difference in density between the truth situation and the assimilated situations.

The second scenario, which has the closing of the bridge as input, performs much better. Due to the sudden change in turn fractions the congestion upstream of the closed bridge is not estimated correctly. It seems that once the congestion has set in, it is hard for the algorithm to estimate the density correctly and has a preference for changing the inflows and turn fractions correctly.

The third scenario performs the best. The east route is estimated almost perfectly, and in the western diversion route the congestion is somewhat underestimated.

Hypothesis 4.2. In non-recurrent conditions, the state based DEnKF estimates the state more accurately than the state based EnKF. In all three scenarios, the DEnKF performs better than the EnKF. In scenario 1, the differences are small: the performance indicators only differ a few percent. The DEnKF scores about 10% better than the EnKF in the second scenario, whereas in the third scenario the performance indicators are more than 30% better.

7.7 Experiment 5: imperfect system model

The previous experiments assumed that a perfect fit exists between the true model that generates the observations and the assimilation model. In a realistic case, this assumption is clearly unfeasible. In this experiment the influence of an imperfect system model on the performance is investigated. In particular, the assimilation model assumes different fundamental relations on the links than the true model.

7.7.1 Goal and hypotheses of experiment 5

In this experiment the critical density and critical speed of the links are varied. This means that the links have different capacity in the true model than in the assimilation model, which influences the traffic flow considerably. One is interested in the performance of the data assimilation.

It is not expected that the density will be estimated exactly right, as the assimilation model considers different density-velocity and density-flow relations. More important is the correct estimation of the main congestion pattern: does the data assimilation estimate the congestion at the right location in space-time, despite the different flow characteristics due to the different link capacities?

Hypothesis 5.1. In an imperfect model context, the main congestion patterns can be estimated reasonably accurate

Hypothesis 5.2. The inclusion of fundamental diagram parameters in the estimation state increases the accuracy of the estimation procedure

7.7.2 Experiment design

Three assimilation scenarios are used, in which the composition of the state is varied. The first scenario uses the same state composition as in the other experiments, i.e. the data assimilation algorithm estimates the cell densities, inflows and turn fractions. The second scenario includes the critical velocity in the state. This way the capacities of the links in the assimilation model are estimated, while maintaining an imperfect fit between the true model and the assimilation model. The third scenario includes both the critical velocity and the critical density in the state. Now all varied parameters are part of the state, so a perfect fit is possible.

In order to limit the time used for this experiment, only 5 different assimilation parameter sets were used for calibration. These 5 sets were based on the parameters that worked well in previous experiments, with some randomized parameters included for the estimation of the fundamental diagram parameters.

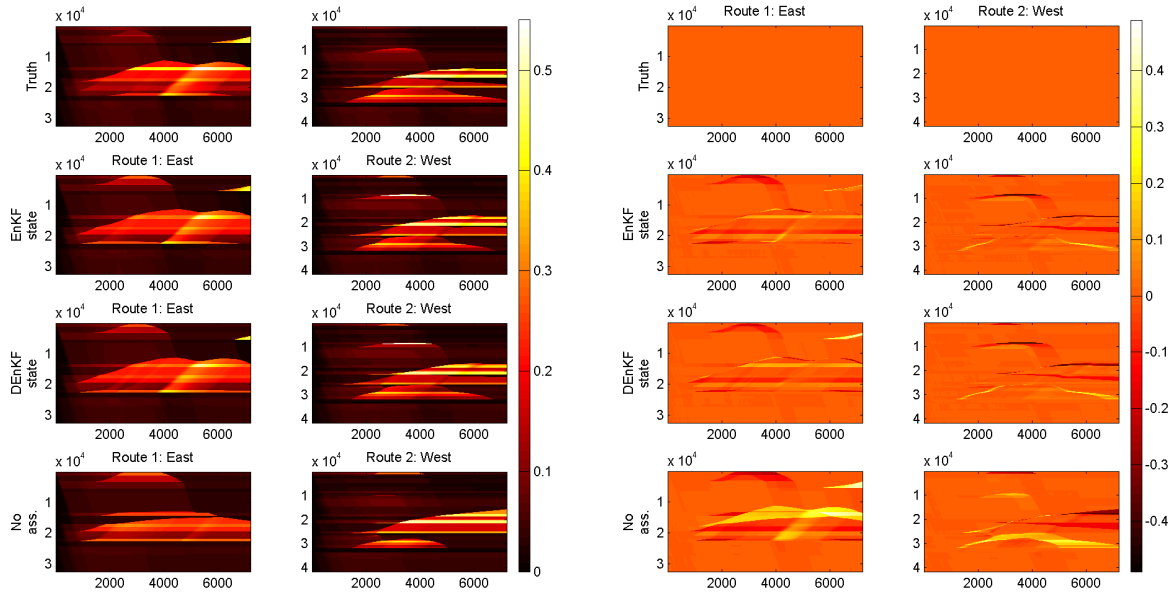
7.7.3 Results experiment 5

Composition state	Method	Set	RMSE K	MAPE K	RMSE V	MAPE V	TRE
No fundamental diagram parameters	EnKF local state	4	0.0283	0.1747	3.3509	0.1533	62394438
	DEnKF local state	5	0.0289	0.1695	3.3941	0.1908	63070574
Critical velocity included	EnKF local state	2	0.0245	0.1361	3.0031	0.1529	53701263
	DEnKF local state	2	0.0191	0.1059	2.4597	0.1041	40672343
Critical velocity and density included	EnKF local state	1	0.0178	0.1008	2.3298	0.0941	46080528
	DEnKF local state	1	0.0130	0.0824	1.8989	0.0580	40378330
No Assimilation		-	0.0565	0.5410	6.1955	0.4688	177779229

Table 7.9: Overview performance of different scenarios. Each scenario adds more estimation freedom to the assimilation model.

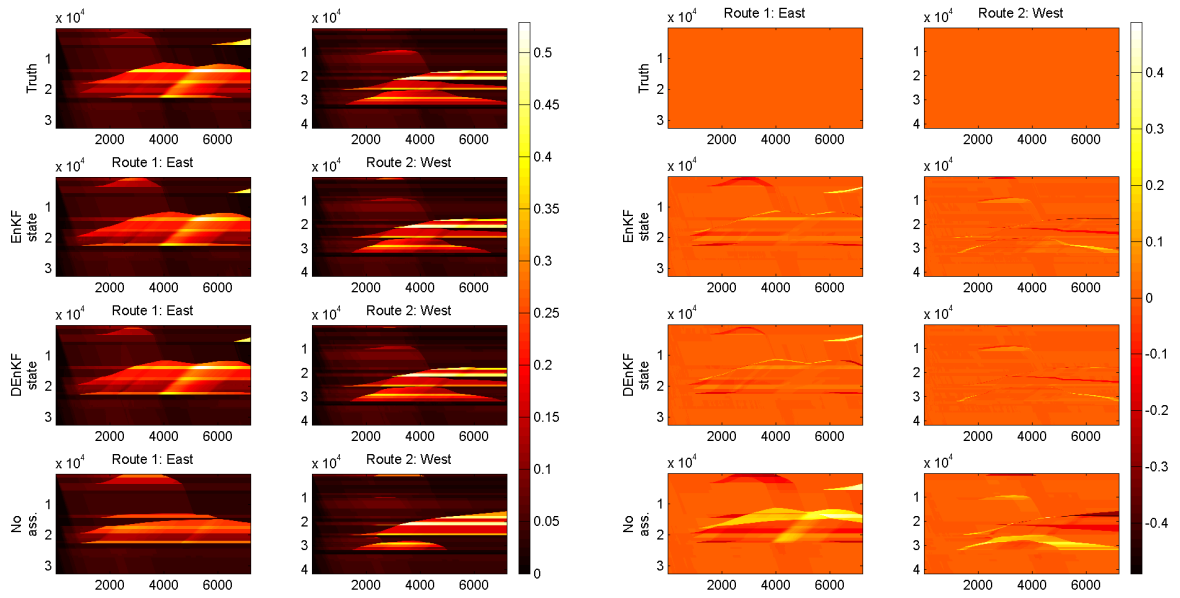
Hypothesis 5.1. In an imperfect model context, the main congestion patterns can be estimated reasonably accurate. In the scenario without capacity estimation, there is quite a large error in the estimated density. Aside from the structural error in density in the congested parts (due to the different observation-state function), also some differences exist in the location of the congestion, as can be seen in figure 7.18. The data assimilation algorithm thus had some trouble in correcting the density to prevent estimating congestion at incorrect locations.

Hypothesis 5.2. The inclusion of fundamental diagram parameters in the estimation state increases the accuracy of the estimation procedure. The estimated state when the capacity estimation was included is much more accurate. Some structural errors in the density of the congestion were present, but the location of the congested area was estimated quite well. The inclusion of both the critical speed as the critical density performed the best.



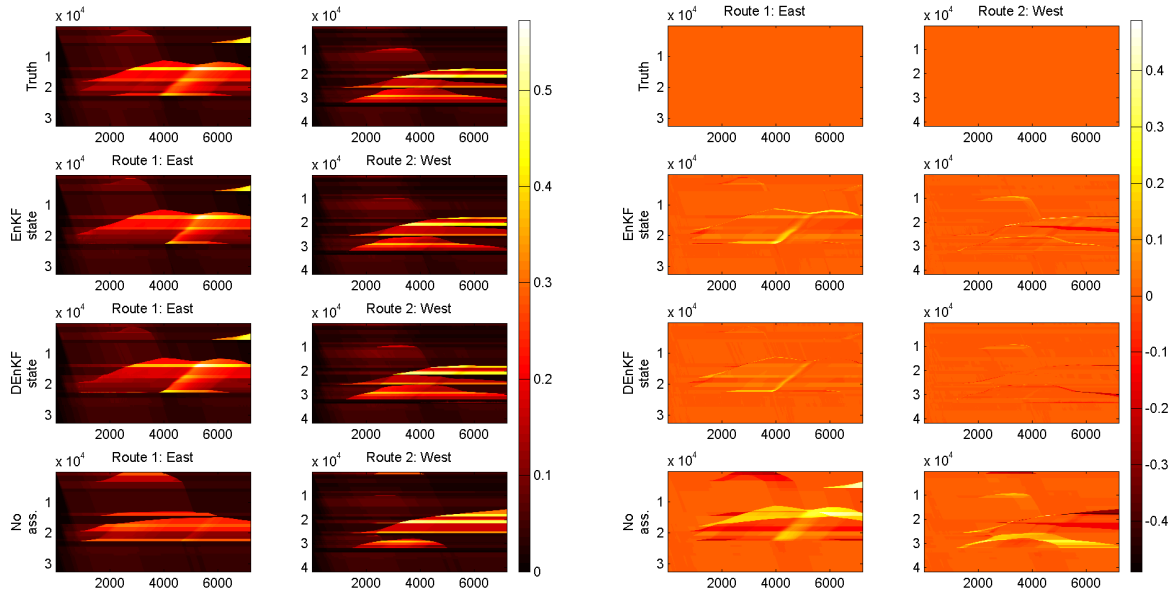
(a) (b)

Figure 7.18: Time-space diagrams of two routes using different assimilation methods. The horizontal axes depict time in seconds and the vertical axes space in meters. Figure (a) describes the density of the cells (in $\frac{\text{veh}}{\text{m}}$); Figure (b) describes the difference in density between the truth situation and the assimilated situations.



(a) (b)

Figure 7.19: Time-space diagrams of two routes using different assimilation methods. The horizontal axes depict time in seconds and the vertical axes space in meters. Figure (a) describes the density of the cells (in $\frac{\text{veh}}{\text{m}}$); Figure (b) describes the difference in density between the truth situation and the assimilated situations.



(a) (b)

Figure 7.20: Time-space diagrams of two routes using different assimilation methods. The horizontal axes depict time in seconds and the vertical axes space in meters. Figure (a) describes the density of the cells (in $\frac{\text{veh}}{\text{m}}$); Figure (b) describes the difference in density between the truth situation and the assimilated situations.

The state localized DEnKF performed better than the state localized EnKF in average. However, the calibration procedure was very roughly done, so a definitive conclusion is hard to draw based on these results.

7.8 Experiment 6: performance of short-term predictions

The main goal of the investigated tool is to get a reasonable prediction in both recurrent and non-recurrent situations. Therefore predictions are evaluated from both the cases in experiment 2 as experiment 4.

7.8.1 Goal and hypotheses of experiment 6

The goal of this experiment is to get some sense in the possibility to get a reasonable prediction result. In order to get a reasonable prediction, it is imperative to estimate the current and future turn fractions and inflow correctly, as the future states are mainly dependent on these variables instead of the current cell densities.

The predictions should work in both recurrent as non-recurrent conditions.

Hypothesis 6.1. It is possible to provide reasonable prediction results in recurrent conditions.

Hypothesis 6.2. It is possible to provide reasonable prediction results in non-recurrent conditions.

Given the previous experiments, the predictions should provide reasonable results in both recurrent and non-recurrent conditions, provided that the input of the length of the external is correct.

Hypothesis 6.3. The state localized DEnKF will provide better prediction results than the state localized EnKF in recurrent conditions.

Hypothesis 6.4. The state localized DEnKF will provide better prediction results than the state localized EnKF in non-recurrent conditions.

In the previous experiments the state localized DEnKF was more accurate than the state localized EnKF. It is hypothesized that the prediction, that is mainly dependent on the correct estimation of the turn fractions and inflows, is also more accurate when the deterministic method is used instead of the traditional stochastic method.

7.8.2 Experiment design

This experiment is based on experiments 2 and 4, which treat the estimation results in recurrent and non-recurrent conditions.

In the recurrent conditions, a prediction is started at $t = 15$min. The performance indicators are computed over different intervals: the used intervals are 5 minutes, 15 minutes, 30 minutes and 60 minutes. In the non-recurrent conditions, the prediction is started $t = 45$min. This is 15 minutes after the malfunctioning of the northbound bridge. It is assumed that the begin and end time of the closing of the bridge is known in advance. The same intervals for the performance indicators are used.

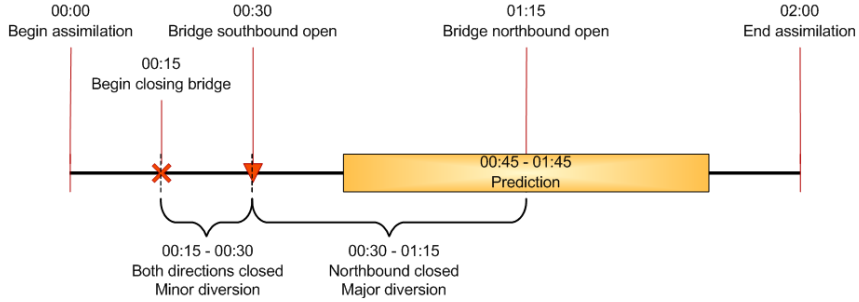


Figure 7.21: Timeline of the bridge closing scenario, including the prediction horizon.

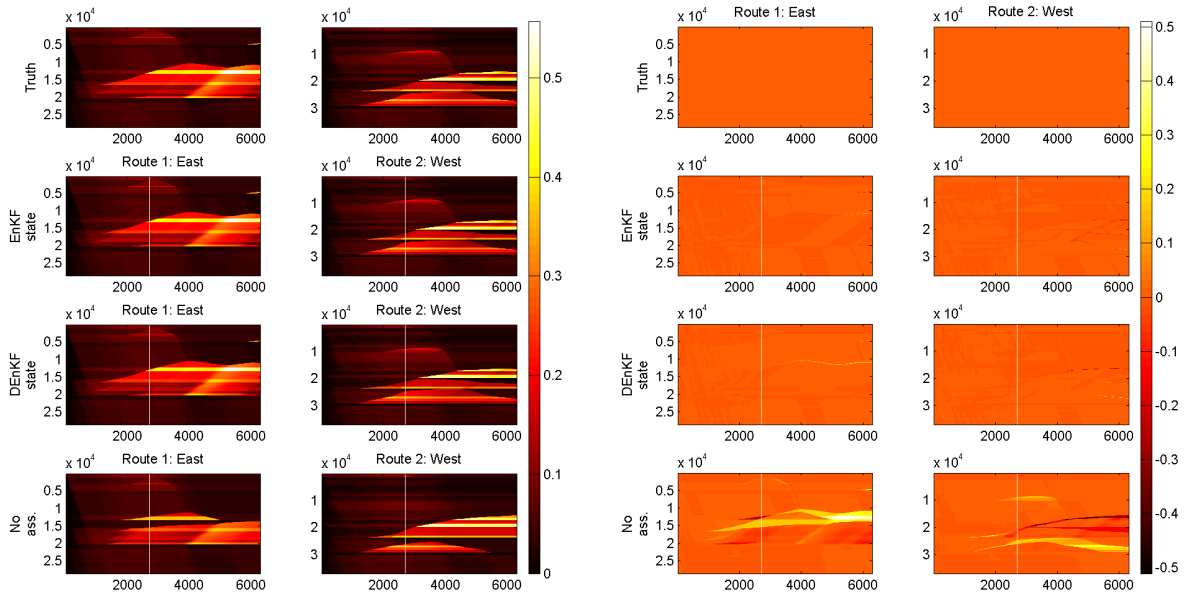
7.8.3 Results of experiment 6

Conditions	Method	RMSE K	MAPE K	RMSE V	MAPE V	TRE
Recurrent	EnKF local state	0.0077	0.0247	0.8989	0.0133	3071263
	DEnKF local state	0.0059	0.0161	0.7258	0.0089	2030633
	No Assimilation	0.0613	0.4970	6.7079	0.5611	102550038
Non-recurrent, input: no additional information	EnKF local state	0.0827	0.2553	7.5415	0.4387	45457102
	DEnKF local state	0.0827	0.2898	7.6492	0.4327	47425832
	No Assimilation	0.0980	0.9341	9.2221	0.6263	81339046
Non-recurrent, input: bridge closing	EnKF local state	0.0544	0.4205	4.9494	0.3381	46578061
	DEnKF local state	0.0508	0.3264	4.8755	0.3522	47505969
	No Assimilation	0.0752	0.8341	6.9248	0.4866	96564293
Non-recurrent, input: bridge closing and route choice estimate	EnKF local state	0.0229	0.1579	2.6364	0.0914	16326713
	DEnKF local state	0.0213	0.1085	2.5781	0.0923	15443111
	No Assimilation	0.0629	0.6179	6.2565	0.4588	82464884

Table 7.10: Overview performance of different scenarios. Each scenario adds more estimation freedom to the assimilation model.

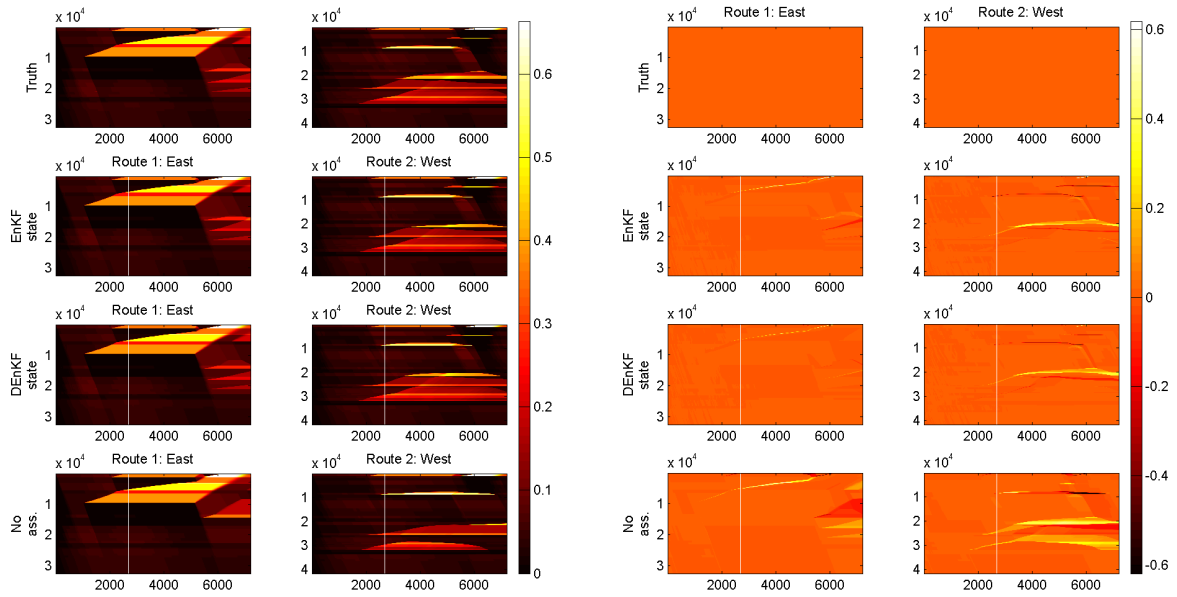
Hypothesis 6.1. It is possible to provide reasonable prediction results in recurrent conditions. The predictions in recurrent conditions were very accurate. This means that the turn fractions and inflows were estimated very well, which resulted in a very accurate prediction of the congestion. This result is very dependent on the input of the boundary condition: the exact begin and end of the peak period was known to the assimilation model

Hypothesis 6.2. It is possible to provide reasonable prediction results in non-recurrent conditions. Just as in the estimation results in experiment 4, the accuracy of the predictions differ on the amount of information put into the assimilation model.



(a) (b)

Figure 7.22: Time-space diagrams of two routes using different assimilation methods in recurrent conditions. The horizontal axes depict time in seconds and the vertical axes space in meters. Figure (a) describes the density of the cells (in $\frac{\text{veh}}{\text{m}}$); Figure (b) describes the difference in density between the truth situation and the assimilated situations.



(a) (b)

Figure 7.23: Time-space diagrams of two routes using different assimilation methods. The used scenario is a scenario of non-recurrent conditions where the closing of the bridge and an approximation of the turn fractions where used as input. The horizontal axes depict time in seconds and the vertical axes space in meters. Figure (a) describes the density of the cells (in $\frac{\text{veh}}{\text{m}}$); Figure (b) describes the difference in density between the truth situation and the assimilated situations.

If the bridge closing isn't put into the assimilation model, the traffic prediction performs very bad. The other scenarios have both quite some error in the prediction. This is most likely caused by a relatively slow response in the estimation of the turn fractions.

Hypothesis 6.3. The state localized DEnKF will provide better prediction results than the state localized EnKF in recurrent conditions. The performance values of the state based localized DEnKF are up to 20% than the results using the state localized EnKF in the tested recurrent conditions. Based on the space-time plots, this difference is very small as both methods perform very well.

Hypothesis 6.4. The state localized DEnKF will provide better prediction results than the state localized EnKF in non-recurrent conditions. As in the recurrent conditions, the deterministic scheme performs slightly better than its stochastic counterpart in non-recurrent conditions.

7.9 Conclusions and discussion experiments

In this section, the six experiments of the previous sections are summarized, and the main conclusions from these experiments are drawn.

In chapter 9, the main assumptions underlying these experiments and the consequences of these assumptions are reviewed.

7.9.1 Main results of experiments

This chapter contains six different experiments.

The first experiment is a verification experiment on a small scale network. The performance of the ensemble based assimilation methods was quite good in terms of estimation and prediction accuracy. The deterministic scheme and the localization had no or very limited influence on the accuracy in this small network.

The second experiment is the base experiment for estimation in a large scale network. In this experiment, it is shown that the localized schemes performed well in reasonable computation time (40× real-time). This experiment indicates that the global methods performed very bad in comparison to the localized methods: no accurate estimation is found using a global method. The deterministic schemes performs slightly better than the traditional schemes.

The third experiment investigates the sensitivity to changes in observation configurations. An observation configuration is comprised of the number of detectors, detector locations, the measured traffic variables and the measurement errors. The results agree with the adage "inclusion of more (reliable) information leads to higher accuracy". Moreover, more theoretical insight is gained about the relation between the shape of the observation

functions and the uncertainty of the state. When the value of the state is uncertain, and thus the ensemble spread is high, the linearised flow-density relation is nearly horizontal. In this horizontal linearisation, a fixed flow uncertainty correspond to very uncertainty density error, which adds almost no information.

The fourth experiment handles the state estimation in non-recurrent conditions. A case is chosen of the closing of a bridge. It is identified that adding of information about the (possible consequences of the) bridge closing is crucial for the estimation accuracy.

The fifth experiment investigates the issues arising with in an imperfect model context. In this experiment, a structural mismatch between the "true model" and the assimilation is introduced by setting the fundamental diagram on the links differently. These changed FD parameters induces quite some error. Inclusion of on-line capacity estimation by the inclusion of the FD parameters in the state increased the accuracy of the state estimation considerably, although the results were not very good yet.

The sixth experiment investigates the prediction capabilities in recurrent and non-recurrent conditions. The accuracy of the predictions is quite good, especially in the recurrent conditions, when the pattern of the future boundary conditions are known. No in-depth analysis of the sensitivity of these boundary conditions is made.

7.9.2 Performance of the three refinements of the EnKF

In this section, the conclusions about the three refinements derived in section 5.3 are drawn on basis of the previous experiments.

Sherman-Woodbury-Morrison formula

From the theoretical analysis the Sherman-Woodbury-Morrison formula (SMW) was found to decrease the computation time of the global methods considerably, without changing the results of the assimilation methods. For the state based localized methods, this (theoretical) effect is far smaller.

In these experiments the SMW formula decreases the computation time using the global methods considerably. In these cases, the median computation time of the correction step is decreased by around 80%. The SMW formula increases the computation time when applied to the state localized methods, and thus isn't useful in these cases. In chapter 6, it was already verified that the SMW implementation doesn't lead to difference in results.

Deterministic EnKF

In the theoretical analysis it is found that the deterministic EnKF by Sakov and Oke (2008) should perform slightly better than the traditional EnKF, as the posterior covariance is analytically approximated instead of statistically. The DEnKF its thus less sensitive to random sampling and a small ensemble. The question remains if the assumption underlying the approximation the DEnKF holds in this model.

In almost all experiments the DEnKF methods perform slightly better than the traditional EnKF methods: the estimation was up to 20% more accurate. The DEnKF doesn't impose a substantial increase on the computation time.

Localization

The localization approaches should increase the accuracy of the assimilation methods, as the spurious correlations between physically distant model elements are omitted. Moreover, the effective ensemble is increased. Two localization approaches are tested in these experiments: the state based localization, where the state elements are localized, and the observation based localization, where the observations are localized. The computation time of the localized approaches should be lower than the computation time of the (unimproved) global methods.

A very large difference in accuracy is found when applied to large scale networks. The global methods ware not useful for application in large scale networks: the global methods score only marginally better than the case where no assimilation method is applied.

The state based localized methods are found to be both (slightly) faster and (slightly) more accurate than the observation based localized methods. Therefore, the choice is made for the state based localized methods instead of the observation based methods.

7.9.3 Comparison computational speed of localized DEnKF

After all these experiments, the question remains if the localized (D)EnKF is a feasible research alternative to the localized Extended Kalman Filter (EKF) by Van Hinsbergen et al. (2012). In table 7.11 the localized EKF of Van Hinsbergen et al. (2012) and the localized (D)EnKF of this research are compared in terms of their computational speed. Although the localized DEnKF is twice as fast, even though the considered network is 2.4× as large in terms of cells and 3.1× as large in terms of measurements, definitive conclusions about the relative computational speed of the localized DEnKF cannot be made yet. The most important reason is that the accuracy of the localized DEnKF is not assessed yet using real data. If the localized DEnKF only can yield similar results as the L-EKF when a large ensemble size is chosen, the computational speed deteriorates. Other reasons for the difficulty in comparing are the different hardware used in the tests and the generalization of the computational speed of the (possibly) suboptimal implementations to the computational speed of the algorithms.

The localized DEnKF of this research provides a feasible research alternative for network-scale state estimation in terms of computational speed compared to the localized EKF. It will be very interesting how the localized DEnKF compares to the localized EKF when real data is used.

	Localized EKF	**Localized (D)EnKF**
Reference	Van Hinsbergen et al. (2012), 75% of detectors used,	*This thesis*, $N = 20$
Hardware	3.0 GHz dual-core, 2 GB RAM	2.6 GHz quad-core i5-3230M, 8 GB RAM
Length network	272 km	264 km
Model time step	5 s	2 s
Number of cells	1911	4656
Number of detectors in network	531	592
Number of used measurements	398	1184
Computational speed	$\approx 20\times$ real-time	$\approx 40\times$ real-time

Table 7.11: Comparison computational speed L-EKF and L-(D)EnKF.

Part III

Conclusions and recommendations

Chapter 8

Conclusions

In this chapter the research questions as posed in the first chapter are answered.

8.1 Main research questions

Question 1: What architecture of a short-term prediction tool will be useful for the current and near-future Dutch operational traffic management practice? In section 3.4 an overview of the proposed architecture is given. Here the main important components are described.

Following subquestions 1a and 1b, the architecture uses a model-based approach for predicting the traffic state. The main functionality is split into two parts: the estimation and prediction components. The estimation component first preprocesses the observational data to reduce the measurement error. The processed data is then used in a Kalman filter approach, together with a macroscopic traffic model. The prediction component uses the estimated state to predict the future traffic state by means of the macroscopic model.

Question 2: Could a monitoring and short-term prediction tool be capable of achieving a sufficient accuracy within the computation time available in a real-time setting? In this research a prototype is developed that tests advanced data assimilation methods using synthetic observational data and a quite simple macroscopic traffic model.

In this setting, the prototype delivered promising results. The prototype was able estimate the traffic state accurately while maintaining a computational speed of 40× real time, if the prototype used a perfect model of the traffic dynamics. When the prototype had imperfect knowledge of the real traffic dynamics, the accuracy dropped. Further research can investigate ways to alleviate the impact of imperfect knowledge of the real traffic dynamics.

8.2 Research subquestions

Question 1a: What functional, performance and stakeholder requirements are imposed on a monitoring and prediction tool? The requirements were derived in section 2.2 using a combination of use cases, previous research and a concise stakeholder analysis. An precise overview of the requirements is given in that section. The most important requirements are also described here.

One of the most important requirements is that the system should be able to estimate the traffic state in both recurrent and non-recurrent situations. The non-recurrent situations are the situations where a monitoring and prediction tool is needed the most as the traffic manager has no or little experience with these situations.

Another important requirement is that it should be able to incorporate control scenarios into the system. This requirement makes it possible to see the influence of control scenarios, which makes the transition to an optimization control approach possible in the future. Moreover, by comparing the network performance in both controlled and uncontrolled conditions, one could get a grasp of the added value of operational traffic management. By making the value of operational traffic management clear, the costs of operational traffic management could be better legitimized.

Question 1b: Which estimation and prediction paradigm suits these requirements best? Two main paradigms were identified: a purely statistical approach (also referred to as a data driven or non-parametric approach) and a simulation based approach (also referred to as model based or parametric approach).

On basis of theoretical considerations, the choice was made for a simulation based approach. This is caused by the fact that the simulation based approach is based on predefined knowledge of traffic behaviour instead of purely historical data. Unseen circumstances such as non-recurrent conditions or control measures thus can be incorporated.

In practice, the statistical methods of the data driven approach can be used in calibrating the simulation model to represent the real traffic situation as good as possible.

Question 2a: Based on the chosen estimation and prediction paradigm, how should the real-time observations be optimally used in the estimation of the traffic situation? In the model-based approach, a data assimilation method combines real-time observations with a traffic model. In the architecture it was found that a Kalman Filter approach would work best in combination with a macroscopic traffic model.

In this thesis, the Ensemble Kalman Filter (EnKF) was analysed as feasible data assimilation tool. In comparison to the Extended Kalman Filter (EKF, commonly used in combination with macroscopic traffic models), the EnKF (commonly used in other fields such as meteorological models) has some theoretical advantages such better handling of non-linearity and lower computational needs.

From literature, three main improvements to the traditional EnKF were identified. The

first improvement is the Sherman-Morrison-Woodbury reformulation, which speeds up the estimation of the state considerably without change in results. The second improvement is avoiding the sampling errors of the (stochastic) EnKF by adopting the deterministic EnKF (DEnKF) by Sakov and Oke (2008). The third improvement is localization of the EnKF, which restricts the relation between model elements (cells and observations) that are physically distant in real-life. The localization improves the accuracy as it removes spurious (i.e. fake) correlations and increases the effective ensemble size.

Question 2b: Based on the chosen estimation and prediction paradigm, is a monitoring and short-term prediction tool capable of achieving a sufficient accuracy faster than real-time using synthetic observations? A prototype was built with the macroscopic LWR model and the EnKF methods. An identical twin experiment was performed, where the observation data was generated by the same model. The case of the road network of Rotterdam is used, which consists of 260 km road.

The local methods performed very well, while maintaining a computational speed of 40× real time. No settings were found where the global methods performed reasonably well. In all experiments tested, the state localized DEnKF performed slightly better than the state localized EnKF.

Question 2c: Based on the chosen estimation and prediction paradigm, how sensitive is a monitoring and short-term prediction tool to imperfect knowledge of the real system? The estimation component is tested using less (reliable) observations, non-recurrent conditions and structural differences in (the fundamental diagram of) the assimilation model compared to the "true model".

Only in an extreme cases of the observation configurations, the performance of the assimilation methods began to seriously deteriorate.

A case of the closing of a bridge was used as case of non-recurrent conditions. The introduction of non-recurrent conditions were quite influential on the performance of assimilation methods. Only if the assimilation model had additional knowledge about the closing of the bridge, the assimilation methods gave reasonable results.

When structural differences exists between the assimilation model and the true model, the estimated state was quite different than the true state. When some sort of capacity estimation is introduced, the performance increases considerably.

Chapter 9

Discussion and recommendations

9.1 Discussion

In this section results and conclusions of this thesis are critically reviewed. This section is split into two parts, which correspond to the two main subjects of this thesis.

9.1.1 Discussion of architecture

1. The assumption is made of a centralized approach using a operational traffic management centre. The future trends can also lead to a more distributed control.
2. The architecture scopes mainly on functional requirements. Other aspects are also important in implementation, for example security.

9.1.2 Discussion of results of prototype experiments

1. The experiment setup assumed a perfect fit of the assimilation model and the true model in terms of model structure. Therefore the accuracy of the assimilation model is far higher than feasible in reality.
2. In the large network experiments, only a limited set of 3 cases was examined. Moreover, the calibration of the assimilation methods were roughly done. Therefore the ability to generalize the experimental results to conclusions about the assimilation methods is relatively hard. However, the experimental results, together with theoretical knowledge, can serve as indication for further research.

   For example the choice between the DEnKF and the traditional EnKF can't be made yet. The limited calibration of the assimilation methods doesn't guarantee that the optimal performance of the methods are compared. However, the performance using only limited calibration possibly indicates the robustness of the methods to the calibration of the parameters.

3. The use of a relatively simple traffic model instead of a complex traffic model possibly increases the accuracy of the data assimilation method, as it is easier to fit the simple model right. This is caused by a smaller number of parameters that need to be fit. In complex traffic models also parameters that are more "non-linear" with respect to the observed speed and flow play a role.

4. The simple model may not emphasize the benefits of the EnKF in comparison to the EKF, as the state space is relatively smooth. In this smooth state space, the strictly local EKF approach performs relatively well. In more non-linear state spaces the non-local approach of the EnKF may perform relatively better.

5. The relatively low performance of the prototype in the imperfect system knowledge cases may be amplified by the simple traffic model. The simple traffic model increases the convergence of the cell densities, as the density is the only parameter in the state that corresponds to a cell. As an example, in congested cells the densities are all equal for all ensemble members as the density of the congested cells are only dependent on the capacity downstream. As the cell densities are all the same for the whole ensemble, the data assimilation algorithm can't change the value of the cell densities.

9.2 Recommendations

The recommendations for the following steps are split into three parts. The first subsection covers the steps to take in further developing the prototype. The second subsection covers future research directions that are not directly related to the development of the current prototype. The last subsection covers recommendations for the practice of operational traffic management.

9.2.1 Further development of prototype

1. *Investigate additive error instead of multiplicative error.* As is discussed, the estimation of density using the simple LWR model and synthetic observations suffers from strong convergence of the cell densities. This implies that a lot of confidence is put on the cell densities by the data assimilation algorithm. This overconfidence in the cell densities leads to the cell densities to be unadaptive to contradicting observations. This can be seen in the several experiments using the imperfect system knowledge: the location of the congestion is estimated wrong. The inclusion of multiplicative errors (by means of covariance inflation) is not sufficient in solving this problem. A change to additive errors may be more successful.

2. *More complex traffic model, multi-class* The prototype uses the (standard) LWR traffic model, which is a fairly simple traffic model. Moreover, this traffic model doesn't incorporate some congestion phenomena such as the emergence of stop-and-go waves and the capacity drop. Using more complex traffic models such as

multi-class models with dynamic pce values can better represent the real traffic dynamics.

3. *Other coordinate systems.* Other coordinate systems such as the Lagrangian approach have some advantages over the used Eulerian coordinate system. An example of these advantages is the less numerical diffusion caused by the discretization.

4. *Handling non-linearity of observations.* In the prototype, the non-linearity of the observation functions is solved by the linearising procedure of the EnKF. Some research propose to incorporate the observations in the system state of the EnKF, which could lead to better results. A disadvantage is that the system state becomes very large, which will slow down the estimation procedure.

9.2.2 Recommendations for further research

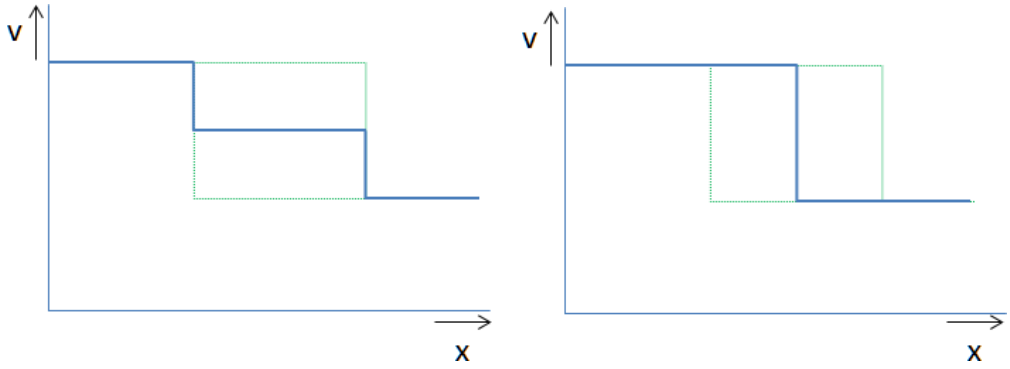
1. *Validation of the prototype.* In this research the prototype is only subjected to synthetic data, generated from an identical macroscopic traffic model. In order to make the following steps to eventual implementation of the traffic state estimation and prediction in an operational traffic management context possible, more research is needed about the behaviour and results using more realistic data. The first step can be to use data generated from microscopic traffic models. The second step could use real data.

2. *Covariance localization.* In this thesis local analysis is used as localization method. Further research can also focus covariance localization, which is another commonly used localization technique. The local analysis technique was chosen mostly

3. *Parameter estimation.* Further research can focus on the way of parameter estimation using the EnKF. One can for example investigate if the inclusion of the fundamental diagram parameters in the state leads to good results. Instead of joint estimation of both parameters and state, one could also use dual estimation, in which the (fundamental diagram) parameters are estimated on a broader spatiotemporal discretization.

4. *Estimation of shockwave position.* The used data assimilation method focuses on the estimation of the *amplitude* of the values in the state vector, for example the density at a certain location. However, one of the main interests is the correct estimation of the *position* of congestion and shockwaves. This is illustrated in figure 9.1. In figure 9.1a the traditional approach is used, where the value of the density is estimated. The mean of the ensemble indicates the mean of the individual cell values. The result is a road stretch where an intermediate velocity is estimated. In 9.1b the position of the congestion is of the shockwave is estimated. Instead of averaging the velocity values on the road stretch where the ensemble members differ, an average trajectory is chosen in terms of the location of the congestion. This second method clearly corresponds better to the physical traffic dynamics.

   This estimation of position can be incorporated by using a Lagrangian coordinate scheme. However, the estimation of position can also be incorporated in the data



(a) (b)

Figure 9.1: These figures plots the velocity of a fictional road stretch. The two green lines indicate the possible trajectories of two ensemble members. The blue line represents the "mean" trajectory of the ensemble.
In figure (a) the mean trajectory is formed by the average of the velocity on every part of the road stretch.
In figure (b) the mean trajectory is formed by the average position of the start of the congestion.

assimilation approach. Further information on adapting the ensemble based Kalman Filters to maintain physical states and coherent features such as shockwaves can be found in Ravela, Emanuel, and McLaughlin (2007), Beezley and Mandel (2008) and Johns and Mandel (2008).

9.2.3 Recommendations for the practice

1. *Integrate state estimation and prediction.* The previous practical experience in traffic state prediction mainly focuses on the prediction part. The estimation component is only treated as necessary part for prediction instead of an interesting component per se. Here it is proposed to integrate the state estimation and prediction. The first reason is the increased trust by the users of the prediction system, when it is shown that the current situation is estimated correctly. The second reason is that due to the future changes in measuring equipment (less cameras, more data from individual vehicles), the implicit state estimation by the traffic operator will become a problem by itself.

2. *Consider other stakeholders such as policy makers.* The previous practical experience with traffic prediction in the Netherlands focused not only on the technical side, but also on the perspective of the traffic operator. This is a good development, as the success of a traffic prediction tool is very dependent on the usability of the tool. However, the traffic operators are not the only stakeholders. Other stakeholders to consider are policy makers that need to approve additional investments in tools for traffic operators. In principle, the prediction tool can be used for "what-if" modeling, and thereby clarify the added value of the actions by the traffic operator.

3. *Use model-based prediction approach.* This research proposes the use of a model-

based prediction approach. This choice was made due to the better handling of non-recurrent conditions and modeling and selecting the appropriate control measures than a purely data-driven approach.