On-line Distributed Prediction and Control for a Large-scale Traffic Network

Yubin Wang

Delft University of Technology

This thesis is a result from a project funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. INFSO-ICT-223844 and by Trinité Automation B.V.

Cover illustration: Yubin Wang and Gang Chen

On-line Distributed Prediction and Control for a Large-scale Traffic Network

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.Ch.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op 10 maart 2015 om 10:00 uur
door

Yubin Wang

Master of Science in Systems and Control,
Technische Universiteit Delft,
geboren te Qianjiang, Hubei Province, China.

Dit proefschrift is goedgekeurd door de promotoren:
Prof. dr. F.M.T. Brazier
Prof. dr. ir. J.H. van Schuppen

Copromotor: dr. J.L.M. Vrancken

Samenstelling promotiecommissie :

Rector Magnificus	voorzitter
Prof. dr. F.M.T. Brazier	Technische Universiteit Delft, promotor
Prof. dr. ir. J.H. van Schuppen	Technische Universiteit Delft, promotor
Dr. J.L.M Vrancken	Technische Universiteit Delft, copromotor
Prof. dr. J. Wu	Tsinghua University
Prof. dr. ir. A.W. Heemink	Technische Universiteit Delft
Prof. dr. ir. E.C. van Berkum	University of Twente
Prof. ir. L.H. Immers	Katholieke Universiteit Leuven
Prof. dr. ir. B. van Arem	Technische Universiteit Delft, reservelid

This thesis is the result of a Ph.D. study carried out on a part-time basis from 2008 to 2014 at Delft University of Technology, Faculty of Technology, Policy and Management, Sections of ICT and of Systems Engineering and Simulation.

TRAIL Thesis Series no. T2015/6, the Netherlands Research School TRAIL

TRAIL
P.O. Box 5017
2600 GA Delft
The Netherlands
Phone: +31 (0) 15 278 6046
E-mail: info@rsTRAIL.nl

ISBN: 978-90-5584-185-1

Copyright © 2015 by Yubin Wang

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the author.

Printed in the Netherlands

从不放弃， 在绝望中寻找希望人生终将辉煌 – 新东方.

”Never give up, hew out of the mountain of despair a stone of hope.”

– Martin Luther King

Preface

After I finished my master degree at Delft Center for Systems and Control (DCSC), Delft University of Technology in 2006, I joined Trinité Automation B.V., that develops and delivers integrated traffic management systems to the traffic control centers in the Netherlands, as a researcher and developer. I was offered a great opportunity to pursue a PhD degree at the moment when Trinité Automation participated in a joint European project, the C4C project, together with TU Delft in May 2008. I really appreciate Frank Ottenhof who is CEO of Trinité Automation and Jos Vrancken who is my daily supervisor of my PhD research for choosing me as a PhD candidate. Additionally, I would like to express my heartfelt thanks to Jos Vrancken for all the help, not only with my PhD research but also with my daily life. You taught me how to carry out scientific research, how to write a good scientific paper and many other things. I still remember how excited I was when my first scientific paper was accepted. Besides, you also helped me with the Dutch language. Thank you again!

I would like to thank my promotor Jan H. van Schuppen who I met during the C4C project. You gave me guidance throughout my PhD research and worked out the mathematics models together with me. Without you, my first journal paper could never have been published. Without you, I could never have finished my PhD. Next, I also would like to thank my other promotor Frances Brazier who gave me many good advices on my thesis. Thanks a lot!

I want to thank Serge Hoogendoorn, Bart de Schutter, Hans van Lint, Chris Tampère, René Boel, Alexander Verbraeck and Andreas Hegyi. Your comments and advice enabled me to pursue this research in the right direction. When I was struggling with my research approach, you always spent your precious time to give me good suggestions. Thanks!

I also want to thank all PhDs: Yufei, Meng, Mohsen, Yilin, Minwei, Ramon, Femke, Thomas, Guus, Victor, Shu, Fangfang, Kassidy and Amir. You helped me a lot with many questions I had during my PhD. The discussions with you were really useful to continue my research. Many of the good ideas in my thesis were inspired by you.

Next, I want to thank my colleagues. Colleagues from Trinité Automation: Marcel Morssink, Marcel Valé, Stephanie, Peter, Paul, Albert Jan, Kresimir, Ferdinand, Henk, Véronique and Sebastian. Colleagues from ATG: Sabien, Janneke, Bas, Tracey, Benny, Charlotte and Olga. Colleagues from Lockheed Martin CFT: Stewart, Hans, Karim, Mike, Duc, Duyet, Hoa, Sabrina, Hillary, Wail, Hao, Cong, Yang, Heather, Hedde, Mark, Julien, Philip, Ubaidullah and Robbert. Without you, my career path hadn't developed so smoothly.

In the end, I would like to thank all my family members who have been always supportive to me. First, I would like to express my sincere gratitude to my parents who encouraged and supported me to study abroad. Second, I would like to thank my wife Ying who gave me suggestions and advice not only for my PhD research but also for my daily work and life. Next, I would like to thank my daughter Candy and my son Denny who brought me endless happiness. After that, I also would like to thank my parents-in-law who came to the Netherlands and helped us to take care of our children. Last but not least, I want to thank Gang Chen who made the cover figure for me.

Thanks to all of you!

Yubin Wang, March 2015

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and background

Transport is an important consideration for global sustainable growth and competitiveness. Among all means of transport, road transport contributes 83.5% of passenger transport and 45.9% of goods transport in 2010. However, road traffic is the most problematic means of transport in most countries. Among the downside of road traffic we find frequent congestion, high environmental pollution and rather high accident rates. Traffic management is a well established means to improve traffic efficiency and to counter the negative effects of road traffic. Road Traffic Control (RTC) is one of the main activities within road traffic management, next to demand management, incident handling and pricing.

From a control engineering point of view, the system schematic representation of an ideal RTC system is shown in figure 1.1. We can see from the figure that the system includes three closely intertwined components: traffic monitoring, traffic prediction and traffic control. Accurate, timely and reliable traffic monitoring including state estimation and data fusion is the basis of performing on-line traffic prediction and applying control strategies to real traffic. Additionally, fast and accurate on-line traffic prediction is crucial to traffic control if the evaluation of control scenarios are needed (of course, traffic control can be performed without traffic prediction). Thus, the overall objective of the system is to control traffic, Traffic monitoring and prediction can be considered to be 'tools' to achieve the goal. In this thesis, the focus is on the development of distributed on-line traffic prediction and control. In addition, traffic monitoring is a topic addressed in this thesis. Traffic monitoring implemented in a distributed environment by combining the existing approaches is briefly introduced in this thesis. The main challenges of this thesis are as the following section.

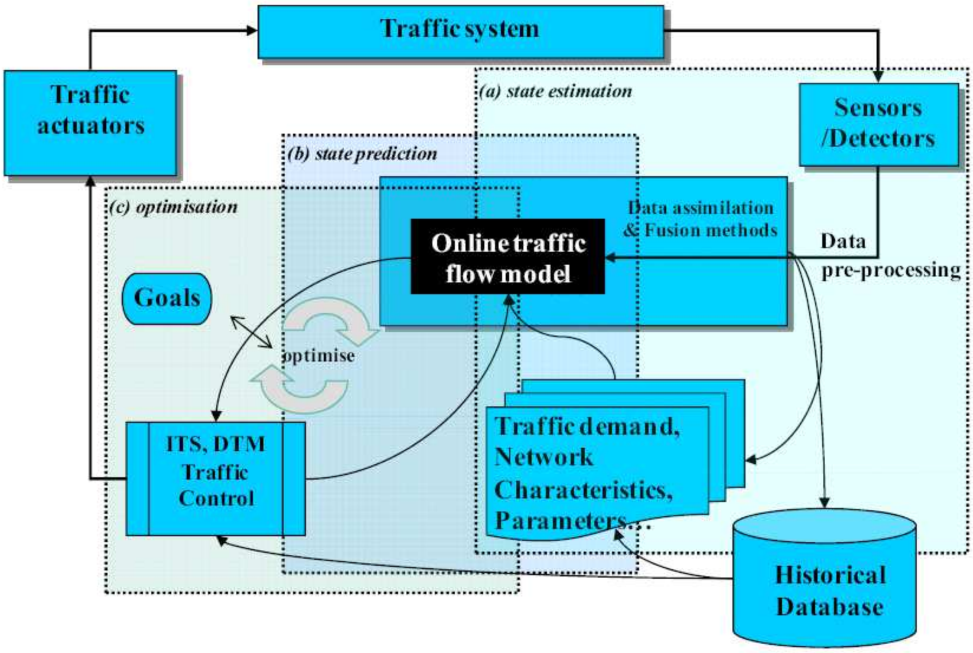


Figure 1.1: Schematic representation of an ideal traffic control system

1.1.1 Road traffic control and its challenges

Networked control

The current status of traffic control in most industrialized countries is (1) that the traffic signal is still the most important measure taken, (2) that a large number of these measures have been deployed, but (3) that the majority of these measures have only a local effect and work largely reactively. Coordinating a number of local measures is still not common, neither is it that traffic control measures can look into the future and respond proactively to upcoming adverse traffic conditions. Thus, one of the challenges of traffic control is to develop multilevel networked control for the coordination of local measures at different levels.

On-line traffic prediction

Regardless which control objectives are used, traffic prediction is needed to evaluate the performance of traffic control, in particular to evaluate the current state of the control measures and to evaluate the performance of one or more control scenarios. Control measures implement a control scenario, such as rerouting, and are themselves implemented using the proper actuators. Another purpose of traffic prediction is to predict upcoming problems in traffic before they actually occur. In general, traffic prediction has four benefits: it is used to achieve optimal traffic control without any risks of impacting the current traffic situation; cost effective (it does not cost much compared with the real experiment); the results of prediction can be studied under different conditions; it can be computed faster than the actual system. However, on-line large-scale traffic prediction must be carry out very fast if one wants to obtain those benefits. Therefore, one of the challenges of traffic prediction addressed in this thesis is to develop a fast prediction algorithm for on-line usage.

1.1.2 Motivating Projects

In The Netherlands, there are five traffic control centers. The Traffic Control Center of the Province of North-Holland is one of the biggest traffic control centers in the Netherlands. Many pilot projects have been carried out in the traffic control center to solve the traffic problems in the area. The research of this thesis is motivated by the following two pilot projects in The Netherlands and a pilot project in Belgium:

The HARS project The aim of the HARS, the Alkmaar Control System (in Dutch, Het Alkmaar Regel Systeem), project is to design and implement a road traffic control system at the belt road of the Dutch city of Alkmaar. HARS operation is challenging in the sense that it uses a large system to control complex behavior (road traffic). The controlled surface includes dozens of actuators and sensors which are distributed in the network.

The FileProof project The aim of the FileProof (FileProof is a Dutch expression, it means 'the experiment with traffic queues') project is to design and implement a traffic control system to reduce traffic jams observed at the Traffic Control Center of the Province of North-Holland in the Netherlands. The area of interest of this traffic control center covers the area around Amsterdam, the area measures about 80 by 50 km. The FileProof project is much more challenging than the HARS project, in the sensor that the controlled surface includes 2462 sensors and 1599 actuators which are distributed in the network of 80 by 50 km.

Large-scale traffic control systems are needed to manage traffic at each of these traffic control centers. The details of the HARS project and the FileProof project are explained in Chapter 8. The details of the Belgium project is explained in Chapter 7.

Furthermore, a prediction algorithm is needed at traffic control centers for on-line prediction of the traffic flow in large-scale road networks, as traffic prediction can be used as a decision support system to evaluate the impact of different scenarios for the purpose of both off-line scenario planning and on-line scenario selection.

1.2 Problem formulation

To solve traffic problems in practice, IT is most often used for ITS. Many ITS companies are trying to solve traffic problems by employing computer engineers who do not have any traffic engineering knowledge. On the other hand, many traffic engineers who are lacking IT skills are working in laboratories and trying to solve traffic problems in simulation environments. However, traffic problems cannot be solved without close cooperation between traffic engineers and computer engineers. There are problems from both the traffic engineering and the computer engineering perspectives.

Centralized control (mainly by using traffic control scenarios) has been used in the Netherlands for the last decade; it is a common approach. However, it is confronted with two main problems, which in recent years became apparent by the development of scenarios for the Alkmaar area in the HARS project and the Amsterdam area in the FileProof project. First of all, due to the size of the network considered and the many states in which this network can be, the number of scenarios needed may easily grow beyond a practically manageable set. As a second problem, even a large set of scenarios still doesn't come close to the many states in which traffic can be, since the control scenarios are developed off-line and correspond to recurring patterns in the traffic state, such as the morning rush hours, or the weekend exodus. They are optimized to a specific traffic pattern, which may differ from real cases. Therefore, there was still a need for local, real-time adaptations of scenarios to local traffic states. In general, centralized control is a common approach but it is may neither produce an optimal traffic flow nor may be feasible for a large-scale network.

On-line prediction is needed to evaluate the performance of traffic control in a real-time application. Centralized prediction has been used in many traffic applications. However, the main drawback is the long computation time when applying it in real-time applications for a large-scale road network. The specification requirements demand that the computations of the predictions should take not more than about 30 sec., preferably less. Another drawback is that the complete prediction has to be computed again when a different control measure is evaluated. However, the part which is not affected by the control measure need not to be computed again. This also leads to a long computation time when a large number of different control measures are evaluated. In general, centralized prediction cannot achieve these practical requirements.

Due to the problems mentioned above, centralized traffic prediction and control are not well able to handle a large-scale road network anymore. In this thesis, we are going to mainly focus on distributed solutions to address these issues.

1.3 Research objectives and scope

The motivation and problem situation of this research are discussed in the previous sections. In this section, we set the research objective and provide a method to tackle this problem as described below.

Research objective to predict and control traffic in a large-scale road network. More precisely, the objective is to solve the traffic problem in a large-scale network by means of distributed traffic control.

To achieve the research objective, the following questions are to be answered in this thesis:

Research Question 1 Is it possible to perform dynamic traffic management in a large-scale network?

Research Question 2 Is it possible to carry out synthesis for control and for prediction of traffic in a large-scale network?

Research Question 3 Is it possible to organize distributed/parallel multilevel computations for control of traffic in a large-scale network?

The first question is to predict and control traffic from a traffic engineering point of view. This is the main question of this thesis to be answered by a theoretical traffic control framework. The second question is to predict and control traffic from a control engineering point of view. Control and system theory provide methods and algorithms to solve the traffic problems of the first and second research questions. The third question is to predict and control traffic from a computer engineering point of view. After the first two questions are answered, multilevel distributed/parallel solution is proposed for a software system. Distributed/parallel means that the computations are carried out on multiple PCs/cores.

To answer the above questions, the following detailed objectives have to be achieved:

- To define a distributed architecture to shorten the computation time and to balance the computational load over the subsystems.
- To develop an efficient algorithm to predict the traffic demand on the boundary of a network.
- To develop a parallel prediction algorithm for the prediction of traffic flow in a large-scale motorway network.
- To design and develop a distributed multilevel control system by using a distributed architecture.
- To develop an estimation model for the purpose of accurate prediction and optimal control.

1.4 Research philosophy and methodology

1.4.1 Research philosophy

The philosophical basis of research is important for numerous reasons. It can assist in understanding notable issues: the interrelation between ontological, epistemological, and methodological levels of inquiry. It is significant in reference to research methodology: (1) to clarify the overall research strategy and to refine and specify the research methods; (2) to enable and assist the evaluation of different methods and avoid inappropriate use; and (3) to help researchers to be creative and innovative in either the selection or the adaptation of methods.

Positivism and interpretivism are common and often seen as opposing views of research philosophy. Their basic differences are listed in Huang (2013). Combining different intervention and research methods can have many benefits in dealing with different dimensions of system problems. The choice of approach may be dependent on the nature of the phenomena of interest, the goal of the research, the level and nature of the questions, experience of researchers and personal beliefs, and practical considerations related to the research environment and the efficient use of resources.

This thesis aims at traffic prediction and control with a focus on mathematical modeling. Positivism is the main philosophical view followed during algorithm and software development. However, systems often involve human behavior (such as driving behavior), and domain experts have their perceptions, values and interests, which will be ignored in this thesis.

1.4.2 Research methodology

The approach of this thesis is according to the standard approach of engineering. It consists of:

- Formulate a problem with a motivation and a limitation of the scope.
- Review and evaluate the drawbacks of the existing approaches in the literature and propose a new approach to fill the gap.
- Formulate an engineering model and a related mathematical model.
- Carry out control synthesis as understood in control theory.
- Evaluate the performance of the proposed approach by methods of analysis and of simulation.
- Formulate comments on the implementation aspects and conclusions of the proposed approach.

1.5 Thesis contributions

The main contributions of this thesis are summarized, distinguishing between scientific contributions and practical contributions.

1.5.1 Scientific contributions

The main scientific contribution of this thesis is to bridge the gap between traffic engineering and control engineering. The traffic engineering problems can be solved using concepts and algorithms of control engineering, in particular of control and system theory. Several main theoretical contributions are listed below.

- This thesis presents a new road network representation which reduces the complexity of a road network and simplifies monitoring, prediction and control of traffic.
- This thesis presents two distributed architectures which are used in two different projects. The architectures can co-exist in one system. The architectures are used to develop a traffic system.
- This thesis presents an adaptive prediction algorithm for traffic flow on the boundary of a network in regular traffic situations based on stochastic control theory. The traffic flow at major motorways on the network boundary and at on-ramps are predicted.
- This thesis presents a novel approach using an asynchronous iterative approach to predict the traffic flow in a large-scale traffic network in a parallel and distributed way. The approach of distributed/parallel computation is promising for other large-scale networks such as energy and rail networks.
- This thesis proves the practical applicability of combining the Treiber and the Kalman filter to estimate missing data and turn fractions of a motorway network.
- This thesis presents a framework of distributed multilevel traffic control by combining a bottom-up control and a top-down control. Distributed traffic control based on a distributed architecture with agents can control traffic at different (more than two) levels. This approach leads the traffic flow to a global optimum by using both the top-town and the bottom-up approaches.

1.5.2 Practical contributions

The main practical contribution of this thesis is to bridge the gap between traffic engineering and computer engineering. The work done in this thesis is challenging in the sense that the algorithms and frameworks were implemented in a real system and they contribute to solve traffic problems by an operational system in practice. In contrast, most of the other works have been made by applying new algorithms and frameworks in a simulation environment. The practical contributions are listed below.

- A software generator is developed to create the new road network representation automatically. It can be used to create a network representation for any country.

- An adaptive prediction algorithm of the traffic demand is designed, developed and tested on traffic flow data of the ring road of Amsterdam. It can be used for traffic demand management.
- An on-line distributed prediction algorithm is designed and developed for the Amsterdam region. It can be easily extended to larger area without major changes.
- Operational software is developed by combining the Treiber filter and the Kalman filter to estimate missing data and turn fractions of a motorway network. It improves the accuracy of traffic state estimation.
- A Bluetooth tracing system is designed and developed to measure the average travel time on motorways. It has been tested in a case study in Belgium. Bluetooth detectors are good alternatives to measure travel time when other sensors are not available. Bluetooth detectors can be widely installed to monitor traffic states for a relatively low cost.
- Distributed traffic control systems are designed, developed and tested in both the Alkmaar area and the Amsterdam area. On average, the control system of the Amsterdam area improves the traffic flow on the ring road for about 16.56% in terms of the total travel time delay. The systems can be easily applied to other regions.

1.6 Outline of the thesis

Figure 1.2 shows the structure of this thesis. This thesis is divided into nine chapters. Chapter 1 is an introduction, which gives background, motivating projects, problem formulation, research scope, research objectives and main contributions.

Chapter 2 sets up a classification framework for distributed traffic prediction and control research. Within this framework, the existing literature is classified into the related categories. This chapter is based on the following papers of which the PhD researcher is first author:

- Wang, Yubin, Jos L. M. Vrancken, and M. S. Soares. A distributed simulator for road network control. In Proceedings of IEEE International Conference on Networking, Sensing and Control, pages 20–25, April 2010.
- Wang, Yubin, J. Vrancken, and M. Soares. Intelligent network traffic control by integrating top-down and bottom-up control. In Control and Decision Conference, 2009. CCDC 2009. Chinese, pages 1557–1562, June 2009.

Chapter 3 presents a new road network representation and two types of distributed architectures. In practice, a generator to create the road network representation automatically is explained in this chapter. The same architectures are used for monitoring,

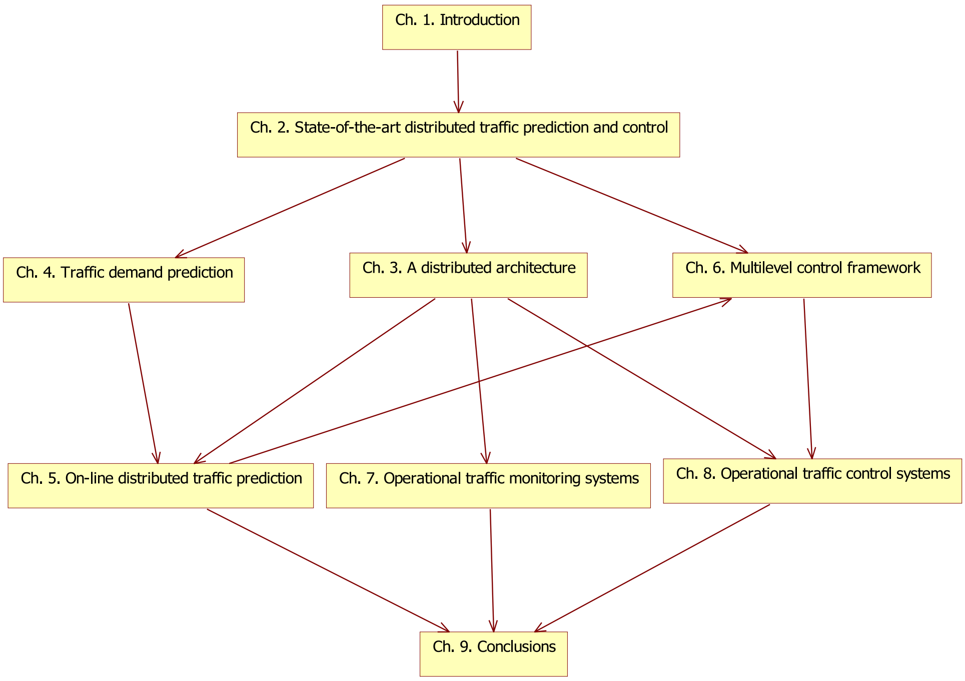

Ch. 1. Introduction
Ch. 2. State-of-the-art distributed traffic prediction and control
Ch. 4. Traffic demand prediction
Ch. 3. A distributed architecture
Ch. 6. Multilevel control framework
Ch. 5. On-line distributed traffic prediction
Ch. 7. Operational traffic monitoring systems
Ch. 8. Operational traffic control systems
Ch. 9. Conclusions

Figure 1.2: Thesis outline

prediction and control of traffic in the Chapter 5, 6 and 7. This chapter is based on the following papers:

- Wang, Yubin, Jos L. M. Vrancken, and M. S. Soares. Road network representation using Dijkstras shortest path algorithm. ITS conference in Sweden 2009, Nov. 2009.

- Wang, Yubin, Jos L. M. Vrancken, and M. S. Soares. Road network representation software generator. In 12th IFAC Symposium on Control in Transportation Systems, pages 190–195, September 2009.

- Wang, Yubin, J. Vrancken, and M. Soares. Intelligent network traffic control by integrating top-down and bottom-up control. In Control and Decision Conference, 2009. CCDC 2009. Chinese, pages 1557–1562, June 2009.

- Wang, Yubin, Jos L. M. Vrancken, and Mohsen Davarynejad. Implementing scenarios coordination for road network traffic control. In Proceedings of the 2010 IEEE International Conference on Systems, Man and Cybernetics, pages 230–234, October 2010.

- Wang, Yubin, J. Vrancken, M. Vale and, and M. Davarynejad. Integration of urban and freeway network control by using a scenario coordination module. In Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, pages 671–676, September 2010.

Chapter 4 presents the design and the performance of an adaptive prediction algorithm for the purpose of short-term traffic flow forecasting at single boundary points which do not have the knowledge of the traffic state upstream from those boundary points. The single boundary point traffic demand prediction plays a key role in supporting a network-based prediction model in Chapter 6. This chapter is based on the following papers:

- Wang, Yubin, J. H. van Schuppen, and Jos L. M. Vrancken. Prediction of traffic flow at the boundary of a motorway network. Intelligent Transportation Systems, IEEE Transactions on, 15(1):214–227, Feb 2014.
- Wang, Yubin, J. H. van Schuppen, and Jos L. M. Vrancken. Adaptive prediction of traffic flow into a motorway network. In Proceedings of the ITS World Congress, October 2011.

Chapter 5 presents a parallel prediction algorithm by using an asynchronous iterative approach and describes the performance of the algorithm for a large-scale motorway network including rings. The distributed prediction will be a tool to evaluate the impact of the traffic control in chapter 6 and 8. This Chapter is based on the following papers:

- Wang, Yubin, J. H. van Schuppen, and Jos L. M. Vrancken. On-line distributed prediction of traffic flow in a large-scale road network. Simulation Modelling Practice and Theory, pages 276–303, September 2014.
- Wang, Yubin, Jos L. M. Vrancken, and M. S. Soares. A distributed simulator for road network control. In Proceedings of IEEE International Conference on Networking, Sensing and Control, pages 20–25, April 2010.
- Wang, Yubin, J. H. van Schuppen, and Jos L. M. Vrancken. On-line distributed prediction of traffic flow in a large-scale traffic network. In Proceedings of the ITS World Congress, October 2012.

Chapter 6 presents the multilevel framework to control traffic in large road networks. The framework explains the traffic control at each level and synthesize a set of control laws to control a multilevel control system for a large-scale road network. The framework offers a scalable approach to network-level traffic management. The framework is the theoretical basis of the operational systems in chapter 8. This Chapter is based on the following papers:

- Vrancken, J.L.M., Yubin Wang and J.H. van Schuppen. QHM: The Quantitative Hierarchical Model for Network-level Traffic Management. In Intelligent Transportation Systems (ITSC), 2013 16th International IEEE Conference on, pages 223–230, September 2013.

Chapter 7 presents two operational traffic monitoring applications to improve the accuracy and reliability of the current traffic measurements. Combination of two filtering techniques are used to supply missing data and estimate turn fraction on motorways. When there are no sensors on motorways, Bluetooth detectors are used to measure the average travel time. The accurate, timely and reliable traffic monitoring plays another key role in supporting a network-based prediction model in chapter 5 as well as traffic control in chapter 6 and 8. This Chapter is based on the following papers:

- Wang, Yubin, Yuan Yuan, and Jos L. M. Vrancken. Traffic parameter estimation on motorway networks by combination of filtering techniques. In Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics, pages 3658–3662, October 2009.
- Wang, Yubin, Jos L. M. Vrancken, and Paul Seidel. Measure travel time by using bluetooth detectors on freeway. In Proceedings of the ITSWorld Congress, October 2011.

Chapter 8 presents two operational applications for distributed traffic control in a large-scale real system and trying to solve traffic problems in practice. The applications are based on the multilevel control framework stated in Chapter 6 and the parallel prediction algorithm stated in Chapter 5. This Chapter is based on the following papers:

- Wang, Yubin, J. Vrancken, and M. Soares. Intelligent network traffic control by integrating top-down and bottom-up control. In Control and Decision Conference, 2009. CCDC 2009. Chinese, pages 1557–1562, june 2009.
- Wang, Yubin, Jos L. M. Vrancken, and Mohsen Davarynejad. Implementing scenarios coordination for road network traffic control. In Proceedings of the 2010 IEEE International Conference on Systems, Man and Cybernetics, pages 230–234, October 2010.
- Wang, Yubin, J. Vrancken, M. Valé and, and M. Davarynejad. Integration of urban and freeway network control by using a scenario coordination module. In Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, pages 671–676, September 2010.
- Wang, Yubin, Jos L. M. Vrancken, F. Ottenhof, and M. Valé. Next generation traffic control in the Netherlands. Traffic Engineering and Control, 3(8):315–319, 2012.
- Wang, Yubin, Jos L. M. Vrancken, and Ying Ma. Evaluation of the scenario coordination module, a traffic network management control system. In Proceedings of the ITS World Congress, October 2012.

Chapter 9 concludes this thesis.

Chapter 2

State of the art in distributed traffic simulation/prediction and control[1]

2.1 Introduction

As has been explained in Chapter 1, distributed traffic control utilizing distributed traffic flow prediction is the main research task of this thesis to achieve the goal of on-line traffic networked control in a large-scale network. Therefore, suitable traffic flow simulation/prediction models as well as appropriate traffic networked control approaches have to be formulated for a real-time environment. For an overview of distributed traffic prediction and traffic networked control approaches, this chapter describes the existing traffic flow prediction models and their distributed solutions in section 2.2 and traffic networked control approaches and their distributed solutions in section 2.3. In the end, a perspective beyond the state of the art is briefly summarized in section 2.4.

2.2 Traffic simulation/prediction models and their distributed solutions

In this section, the existing traffic flow simulation/prediction models are summarized in section 2.2.1. Then the general techniques for the distributed parallel simulation/prediction are explained in section 2.2.2. In the end, the distributed parallel simulation/prediction techniques for the traffic domain are described in section 2.2.3.

2.2.1 Traffic flow simulation/prediction models

Traffic flow simulation/prediction models describe the interaction of vehicles with the infrastructure in a simulation environment. The purposes of traffic simulation/predic-

[1] This chapter is based on my publications (Wang et al., 2010a, 2009a)

tion for an on-line traffic control system are:

- to predict upcoming problems in traffic before they actually occur;
- to test sets of control measures for their effectiveness with respect to a specified purpose.

The two main purposes imply that a traffic simulation/prediction in an operational control setting has to comply with strict real-time requirements, simply because a prediction is of value only if it is known earlier than the predicted event.

Traffic simulation/prediction models have been developed since the early twentieth century. (van Wageningen-Kessels et al., 2012; Zegeye, 2011) reviewed traffic models intensively. In this thesis, a short summary of the models is described. Based on their level of detail, they can be classified into three types: microscopic, macroscopic and mesoscopic traffic simulation/prediction models.

Microscopic traffic simulation/prediction models

Microscopic models describe the behavior of individual vehicles. In the models, lane changes, the inter-vehicle distance, and the effect of neighbouring vehicles on a vehicle are described. The main advantage of microscopic traffic flow models is that the behavior of the drivers and vehicles are described in very much detail. However, the main limitation of microscopic models is that they require a large memory size and they are very slow when used for a large traffic network (Mahmassani et al., 1990; Nagel & Schleicher, 1994).

Microscopic models can be classified into five categories based on their conceptual approaches (Hoogendoorn & Bovy, 2001b): car-following (Gipps, 1981; Chandler et al., 1958; Gazis et al., 1961; Kerner & Klenov, 2006; Treiber et al., 2000), microscopic simulation (Hoogendoorn & Bovy, 2001b; Vermijs et al., 1995), submicroscopic simulation (Hoogendoorn & Bovy, 2001b), cellular automaton (Cremer & Ludwig, 1986; Kai Nagel & Michael Schreckenberg, 1992; Helbing & Schreckenberg, 1999), and particle models (Hoogendoorn & Bovy, 2001b; Marinica & Boel, 2011). Most microscopic models are car-following and lane changing models.

Macroscopic traffic simulation/prediction models

Macroscopic models represent traffic in terms of aggregate variables such as flow and density. In the models the simulation time and memory requirements mainly depend on the size of the spatio-temporal discretization, but not on individual cars. Therefore, macroscopic traffic flow models are suitable for real-time traffic simulations. Most macroscopic models are developed under the assumption that traffic dynamics are comparable to fluid dynamics, resulting in models with a limited number of recursions that are relatively easy to compute (Hoogendoorn & Bovy, 2001b). Furthermore, macroscopic models can accurately describe the average traffic dynamics of freeway traffic (van Lint et al., 2008).

The first macroscopic traffic model called the LWR model is proposed in (Richards, 1956; Lighthill & Whitham, 1955). The LWR model is developed further to achieve finer and more accurate traffic flow models in (Kotsialos et al., 2002a; van Lint et al., 2008; van Wageningen-Kessels et al., 2011).

Mesoscopic traffic simulation/prediction models

Mesoscopic traffic flow models are developed to fill the gap between microscopic models and macroscopic models. In the models the vehicle or the driver behavior is not described individually, but in aggregate variables. However, these models become more complicated to simulate and calibrate than their corresponding microscopic or macroscopic versions, because the mesoscopic models combine some of the microscopic characteristics to macroscopic models or vice versa.

Mesoscopic traffic flow models can be subdivided into the following three categories Hoogendoorn & Bovy (2001b): headway distribution models Buckley (1968); Branston (1976), cluster models (Kühne et al., 2002), and the gas-kinetic continuum models (Paveri-Fontana, 1975; Hoogendoorn & Bovy, 2001a). The most popular mesoscopic models are gas-kinetic continuum models which are also the oldest models among the three categories.

In general, as we described above, the main limitation of microscopic and mesoscope models is that the computation is much more complex and very slow for a large scale traffic network. So, macroscopic traffic models are used in this thesis.

2.2.2 General distributed parallel simulation/prediction

In order to use traffic simulation/prediction for an on-line traffic control purpose, a short computation time is crucial for the real-time traffic control. In order to achieve a fast computation, a distributed parallel approach of traffic simulation/prediction is a well-known and efficient approach for a large-scale simulation. A distributed parallel approach allows simulation/prediction computing on multiple PC at the same time, such that the computation time is dramatically reduced.

General distributed parallel simulation is classified into four categories based on the time window: Conservative, Optimistic, Relaxed synchronization and Mixed-mode.

Conservative

In general, this approach always ensures safe timestamp-ordered processing of simulation events within each Local Processor (LP). In other words, an LP does not execute an event until it can guarantee that no event with a smaller time stamp will later be received by that LP. Three techniques in literature are used to improve the conservative parallel simulation.

Lookahead approach (Perumalla, 2006) In this approach, events that are beyond the next lookahead window are blocked until the window advances, sufficiently far to cover those events.

Bounded lag algorithm (Lubachevsky et al., 1988) In physical systems there is always some delay between the time when one part of the system does something, and time when other parts of the system realize that something was done, a simulation system is realized that avoids all blocking and advances the simulation time in an efficient manner. The efficiency is achieved by each node independently evaluating for itself a time interval that is not 'at risk' and simulating all events that are within that evaluated time interval.

Critical Channel (Xiao et al., 1999) The method focuses on task scheduling. It is designed for describing and taking advantage of the inherent locality of event execution in conservative simulations. Benefits include better cache performance due to immediate reuse of event buffers across causal chain of events and improved spatial locality due to reflection of LP inter-dependence on event processing patterns.

However, the lookahead time window or the 'no risk' time interval is very hard to extract in complex applications, as it tends to be implicitly defined in the source code interdependencies.

Optimistic

In general, this approach avoids blocked waiting by optimistically processing the events beyond the lookahead window. When several events are later detected to have been processed in an incorrect order, the system invokes compensation code such as state restoration or reverse computation. The following techniques described in the literature are used to improve the conservative parallel simulation.

Time Warp (Jefferson, 1985) This is a well-known parallel discrete synchronization protocol that detects out-of-order executions of events as they occur, and recovers using a rollback mechanism. The main drawback is that the algorithm is based on overoptimistic assumptions. The execution of events may suffer from excessive rollbacks which induces memory and communication overheads as sources of performance inefficiencies.

Local adaptive protocol (Hamnes & Tripathi, 1994) The core of the algorithm is how events are chosen for execution. The algorithm uses simple locally maintained statistical data. An empty channel is blocked if the increment in simulated time between the last event processed and the time of the event to be processed exceeds the average interarrival time. It allows each node to locally decide on a per channel basis the best method of handling events.

Adaptive time warp (Panesar & Fujimoto, 1997; Ferscha & Luthi, 1995) The number of scheduled uncommitted messages for each processor are estimated and used to define a moving window that defines an upper bound on the number of uncommitted messages an LP can have scheduled at a considered time. This window is expanded so that the process may progress up to the next commit point.

Filtered rollback algorithm (Lubachevsky et al., 1989) The estimation of the 'no risk' time interval in Lubachevsky et al. (1988) may be difficult, undesirable or impossible. 'Filtered rollback' is proposed to improve the concurrence of the simulation by selecting a larger time interval than the 'no risk' time interval.

Efficient state saving (Perumalla, 2006) One of the common approaches for rolling back incorrect computations is based on checkpointing, also called state saving. Two state saving techniques are copy state saving (CSS) and incremental state saving (ISS). Two main drawbacks of state saving techniques: (1) a memory copy overhead during forward computation; (2) memory consumption beyond that of conservative execution.

Reverse computation (Perumalla, 2006) The values of state variables are not saved for rollback support. Instead, as and when rollback is initiated, a perfect inverse of event computation is invoked that serves to recover the modified state values to their original values. In other words, the overwritten state values are reconstructed by executing the forward code backwards.

However, optimistic approaches might cause a lot of rollback overhead which can degrade the performance of parallel simulation/prediction. Sometimes the overhead makes the performance of optimistic approaches even worse than the performance of conservative approaches

Relaxed synchronization

This approach relaxes the constraint that events be strictly processed in time stamp order Perumalla (2006). For example, it might be deemed acceptable to process two events out of order if their time stamps are close enough. This approach offers the potential of providing a simplified approach to synchronization, but without the lookahead constraints that plague conservative execution.

Mixed-mode

This approach combines elements of the previous three Perumalla (2006). For example, sometimes it might help to have some parts of the application execute optimistically ahead, while other parts execute conservatively.

All four types of the general distributed parallel simulation/prediction approaches are based on a time window which means events are processed in the sequence of their time stamps. Therefore, these approaches have a strong limitation on simulation concurrence which makes the simulation be determined by the slowest LP.

2.2.3 Recent advances in distributed parallel simulation/prediction

In order to solve the problem mentioned above, two recent advances are described in this section.

State Matching

Another approach called state matching is described in Fujimoto (2000) for time-parallel simulation. In this approach, simulation in each time interval can be performed concurrently, if the simulation of a time interval starts from an initial state that is inconsistent with the final state of the previous interval, correcting these afterwards by the use of fix-up computations. Instead of an exact solution to the problem, approximate state matching Kiesling & Pohl (2004); Kiesling & Luthi (2005) is a heuristic approach that accepts deviations in final and initial states of adjacent intervals while guaranteeing the quality of the simulation.

In this approach, a number of iterations are needed in order to have consistent states at two adjacent intervals. However, convergence analysis is not addressed in these papers. Therefore, convergence is not guaranteed. In addition, the iterations might lead to a lot of overhead which can degrade the performance of parallel simulation.

Gluing algorithm

A gluing algorithm is proposed in Wang et al. (2003); Nakhla et al. (2010) to solve the convergence problem for distributed simulation of physical systems and power grid systems. At each time step, each subsystem simulate in parallel by guessing the initial states on the boundary points. Then a Newton-Raphson method is used to fix-up the mismatch of the initial states. By using this approach, the convergence is guaranteed. However, a large number of iterations might be needed if the guessed initial states are far away from the actual states. In Robbins & Zavala (2011), a parallel Newton method is proposed to reduce 75% of iterations. The method can be interpreted as a Newton-Raphson method applied to a system in a entire simulation horizon instead of each time step. In other words, the parallel Newton method solves the convergence problem for the entire simulation horizon of a system simultaneously. As a result, the number of iterations can be significantly reduced.

However, the parallel Newton method needs a centralized cooridinator to compute a matrix inverse which is also cost expensive. In Hoogendoorn et al. (2003), an asynchronous iterative approach has been developed to predict the conditions in the network for the purpose of real-time scenario evaluation. However, the paper only gives a conclusion (without more detailed explanation and provement) that a few iterations (less than 10) are needed to achieve the convergence.

2.2.4 Distributed traffic simulation/prediction

In this section, the existing distributed traffic simulation/prediction applications are classified into two categories: distributed microscopic traffic simulation and distributed macroscopic traffic simulation.

Distributed microscopic traffic simulation

Simulation packages like ParamGrid Klefstad et al. (2005), AIMSUN Barcelo et al. (1998), Paramics Cameron & Duncan (1996), and TRANSIMS Rickert & Nagel (2001) ;Nagel & Rickert (2001) are distributed, using various forms of middleware. Raymond Klefstad Klefstad et al. (2005) used CORBA as communication middleware in distributed simulation. Ulrich Klein Klein et al. (1998) applied traffic simulation on HLA to realize the interoperation of simulation systems. A parallel large scale microscopic traffic simulation system called PMTS is presented in Dai et al. (2010). In PMTS the concept of divide-and-conquer is applied: using MPI (Message Passing Interface) as communication middleware the simulation traffic network is divided into sub-networks, and then each sub-network is run in parallel on a cluster of computers connected by high-speed Ethernet transferring vehicles across sub-networks when necessary. In Potuzak (2008), an adaptation of the Java Urban Traffic Simulator (JUTS) for distributed computing environment is described to reduce the interprocess communication which is one of the main bottlenecks of any distributed application. In Lee & Chandrasekar (2002), a method to use a simulation program for running parallel simulation has been described and demonstrated. The main idea is to divide the network into several regions and simulate under different instances of the program simultaneously, allowing transfer of vehicles at the boundaries of regions. Thulasidasan & Eidenbenz (2009) presents FastTrans - a parallel, distributed-memory simulator for transportation networks that uses a queue-based event driven approach to traffic micro-simulation.

Distributed macroscopic traffic simulation

In Johnston & Chronopoulos (1999), 1-step and 2-step algorithms have been presented to solve the highway traffic simulation equations with fewer inter-processor communication. In Kwon et al. (2000), a PC-based distributed macroscopic traffic simulation methodology has been presented to apply to real-time estimation of time-variant parameters in freeway traffic flow. The method uses the interprocess communication procedures currently available under Windows NT and adopts two types of distributed computing architectures, i.e., a distributed-memory structure with a network of PCs communicating through Named Pipe and a multithread-based shared-memory structure with a multiprocessor PC.

2.3 Traffic networked control approaches and their distributed solutions

The literature offers an abundance of approaches to traffic networked control. In this section, the general idea of traffic networked control approaches both for freeway networks and urban networks are summarized and distributed solutions of these approaches are described. They are classified into two categories: feedback control approaches and predictive control approaches. Feedback control approaches react to the current traffic pattern and solve the traffic problems after they actually happened. However, predictive control approaches consider the future traffic pattern based on traffic prediction models and prevent traffic jam in a pro-active way. Feedback control approaches include scenario-based traffic control, case-based traffic control and fuzzy rule based traffic control. Predictive control approaches include optimal control, model predictive control and Receding-Horizon Parametrized Control. In principle, all above approaches can be implemented in both a centralized and a distributed way.

2.3.1 Feedback control approaches

Scenario based traffic control

The best known approach Rathi (1988) in network control is the one characterized by traffic management centers applying so-called scenarios in response to current and predicted traffic states of the road network. Scenarios are a coherent set of local measures for a road network. They are developed off-line and correspond to recurring patterns in the traffic state, such as the morning rush hour or the weekend exodus. It is a common, but limited, way of controlling the overall traffic performance in a network.

Case-based traffic control

Case-based traffic control solves a problem using the knowledge that was gained from previous experience in similar situations Aamodt & Plaza (1994); Ritchie (1990). This technique stores the new solution in a database once it has solved a new problem. A disadvantage of this approach is that it might not be clear what should be done in case of a problem which is not yet presented in the database. However, new cases could be added on-line to deal with this disadvantage.

Fuzzy rule based traffic control

Fuzzy rule based traffic control can be used when accurate information of the traffic model is difficult to obtain or is not available Krause & von Altrock (1997). In this approach, several fuzzy sets can be defined such as fuzzy sets for local speed, local traffic flow, queue occupancy, metering rate, and local occupancy based on relevant variables and their corresponding traffic situations. A main difference between this approach and the case-based approach is that the case-based approach usually has a fixed solution, whereas the fuzzy rule based approach would apply a combination of traffic control measures for each case in the case base Hegyi et al. (2001). One of main difficulties of a fuzzy rule based traffic control is the selection of appropriate membership functions in order to achieve the best combination of different traffic control measures.

2.3.2 Predictive control approaches

Optimal control

The main idea of optimal control is to find that control measures of the entire network for the considered future horizon which minimizes the cost function based on a network model. It can not only coordinate the control measures on different space locations and different time points in the future, but also coordinate different types of control measures.

However, one of the big challenges of applying optimal control is to find an efficient algorithm to solve the complex optimization problem which in general takes a long computation time. It solves the optimization problem based on the approximation of the future network disturbances, which can be inaccurate, or even be the opposite of reality when unpredictable events occur.

AMOC (Advanced Motorway Optimal Control) Kotsialos & Papageorgiou (2004a) and OASIS (Optimal Advanced System for Integrated Strategies) are two control software tools based on optimal control theory. A feasible-direction nonlinear optimization method Kotsialos & Papageorgiou (2004b); Kotsialos et al. (2002b) is proposed to solve the optimization problem efficiently. To avoid the inherent drawbacks of an open-loop control, Traffic-responsive Urban Control (TUC) Dinopoulou et al. (2006); Kosmatopoulos et al. (2006) is developed based on a store-and-forward model. Instead of optimizing the control inputs (i.e. green times), TUC optimizes the linear multivariable feedback regulator off-line.

Model predictive control

Model Predictive Control (MPC) Camacho & Bordons (1997); Mayne et al. (2000) is a model-based control approach that implements and repeats optimal control in a rolling horizon way. This means that, in each control step, only the first control time step of the optimal control sequence is implemented, subsequently the horizon is shifted one sample and the optimization is restarted again with new information of the measurements. The optimization is calculated based on the prediction model of the process and of disturbances.

MPC preserves several advantages of optimal control. Moreover, it takes the effect of the control inputs on the future system states, that it is able to take both equality and inequality non-linear constraints of the manipulated and controlled variables into account, and that it can be used for non-linear systems Zegeye (2011). Another advantage is that MPC is a closed-loop control instead of an open-loop control.

However, as for optimal control, the real-time computation complexity is one of the big challenges for implementing MPC controllers to traffic networks in practice. To solve this problem, many techniques are proposed to make the real-time computation feasible Bemporad et al. (2002); Goulart et al. (2006); Kothare et al. (1996); Mayne et al. (2005). Recently, a model-predictive hierarchical control approach is proposed in Kotsialos et al. (2005); Ghods & Rahimi-Kian (2008); Lin (2011).

Receding-Horizon Parameterized Control

As we mentioned above, the computation time exponentially increases as the size of the road network or the prediction horizon increases. Another interesting approach that can reduce the computation time considerably is Receding Horizon Parameterized Control (RHPC). In this approach, the controller optimizes a set of parameters instead of optimizing a sequence of control inputs. The parameters are optimized in such a way that a given performance criterion is improved in a receding horizon fashion. To do so, the control inputs are described using certain control laws that are dictated by the values of the parameters Zegeye (2011).

RHPC combines the advantages of conventional MPC (i.e., prediction, adaptation, and handling constraints, multi-objective criteria, and non-linear models) and the advantages of state feedback controllers (i.e., faster computation speed and easier implementation).

However, depending on the parametrization the performance of the RHPC approach is then most often reduced.

2.3.3 Distributed traffic control

In general, distributed traffic control can be developed to avoid the exponential increase of the computational complexity for the centralized traffic control, when the size of the network is getting larger and larger. In this section, the existing distributed traffic control approaches are summarized.

An agent-based approach can be found in Adler et al. (2005), in which traffic is managed by negotiations between the different actors involved. The possibility of combining both distributed and generic control is analyzed in Immers et al. (2008a). An approach to traffic control based on a prediction model can be found in Hegyi (2004). A new adaptive multi-agent control of a network equipped with traffic signals is presented in Katwijk (2008). Recently, a model-predictive hierarchical control approach is proposed in Kotsialos et al. (2005); Ghods & Rahimi-Kian (2008); Lin (2011).

2.4 Beyond the state of the art

The existing approaches for both distributed simulation/prediction and control have been reviewed. In this section, the main drawbacks of the existing approaches will be discussed and the benefits of our proposed approaches will be briefly introduced.

2.4.1 Distributed traffic prediction

All the above traffic simulation/prediction applications are based on the parallel techniques in section 2.2.2, so that the applications inherit the drawbacks of the techniques. The main drawback as we mentioned in section 2.2.2 is that the techniques have a strong limitation on simulation concurrence which leads the simulation determined by the slowest LP. Thus, in this thesis, the asynchronous iterative approach in section 2.2.3 combined with adaptive predictions at boundary points is explained in more detail and demonstrated in a case study of both a linear model and a non-linear model. More precisely, using as the initial guess for the inputs of the subnetworks predictions from a previous time step or of adaptive predictions. Then, the drawbacks of synchronized approaches can be prevented. The advantages of the proposed approach are:

- Computation speed: The proposed parallel simulation does not depend on the computation speed of the slowest agent in the sense that it can already have computation results before finishing the computation of the slowest agent, although one iteration might need to refine the simulation/prediction results later.

However, the computation speed of the traditional parallel simulation depends on the simulation speed of the slowest agent (task).

- Loss of communication: The proposed parallel simulation can advance the simulation based on the initial guess (whether from adaptive filter or from history data on the boundary points). Although the simulation might not be accurate, the initial guess on the boundary points can still give the best possible inputs to advance the simulation. Therefore, the simulation can be performed even if one agent is not performing or stuck in its simulation. However, the traditional parallel simulation is not able to advance the simulation if communication between agents is lost.

- Control measures evaluation: The proposed parallel simulation only needs to simulate agents which are affected by the control measures. However, full simulation of all agents has to be performed again for traditional parallel simulation.

- Scalability: When the network is extended further to the Netherlands' road network or the European road network, the proposed parallel simulation is scalable. The total simulation speed does not change too much when the size of the network increases. However, the simulation speed decreases dramatically with the size of the network for a centralized simulation.

2.4.2 Distributed traffic control

All the above traffic control approaches have a common element - an agent which is trying to solve problems that are difficult or impossible individually. Agents are autonomous software entities capable of communication and cooperation Wooldridge (2009). However, it is hard or even impossible to reach a system optimum. Furthermore, the overview control from the traffic control center is lost. To overcome the problem, the multilevel traffic control in a large scale road network is proposed. In the proposed control algorithm, the traffic problem is solved at different control levels by using different agents in a multilevel architecture. In this multilevel architecture, the following two approaches can co-exist: (1) Top-down; (2) Bottom-up. Advantages of the proposed approach are:

- A policy dependent control algorithm: Different priorities can be assigned based on a policy at different locations.

- Scalable: The control algorithm can be easily extended to any size of road networks.

- Reduced the complexity of network control: Control is done at different levels to distribute complexity of control over the network.

Chapter 3

A distributed architecture based on a new road network representation[1]

3.1 Introduction

Much computation is needed for traffic monitoring, simulation and control of a large-scale road network. A distributed approach of computation based on a distributed architecture is needed to solve large computational problems. Moreover, the traffic measures are distributed with sensors and actuators spread over various locations. The distributed operation makes the system more robust in case of failures. If one machine crashes, the remainder of the system can often function normally. A distributed architecture improves the scalability, allowing the system to be extended.

Furthermore, traffic monitoring, simulation and control are complex, among other reasons because of the complexity of road networks, as illustrated in figure 3.1 by the motorway network in the province of North-Holland of the Netherlands. To make this complexity manageable, proper representation and visualization of road networks are important. Road network representation is necessary for three main objectives: traffic monitoring, traffic simulation and traffic control. The important goal of network representation is to simplify road networks such that these three functions are not impaired, but also benefit from the reduction of complexity.

This chapter is organized as follows. Section 3.2 presents a new road network representation to reduce the complexity of road networks. Section 3.3 describes a software generator to generate the new road network representation automatically. Section 3.4 presents distributed system architectures based on the representation. Section 3.5 introduces a distributed middleware to further simplify the implementation of traffic system by hiding the complexity of low-level communication. Concluding remarks are made in Section 3.6.

[1] This chapter is based on my publications Wang et al. (2009b,c,a, 2010c,b)

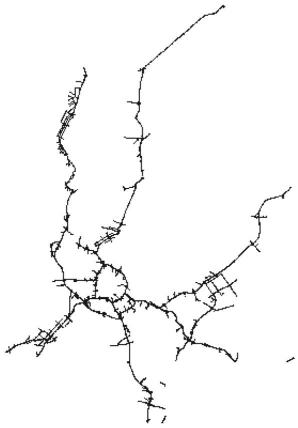


Figure 3.1: The complex road networks in the Netherlands.

3.2 A new road network representation

In Sakai & Nagao (1969), the most common representation for road networks is by a list of links and nodes. Links are road segments with one driving direction and nodes are intersections of roads connecting the links. Most of applications use the common representation, such as described in Wuest & Mioc (2007); van Lint et al. (2008); Mark et al. (2004); Xiong (1992). The detailed Dutch road network database *Nationaal Wegenbestand* (NWB) NWB (2002) is also represented in this way using shape files ESRI (1998). The NWB is a digital geographical file which contains virtually all the roads in the Netherlands. It consists of three types of files: .shp, .dbf and .shx files. Shape files exist for many countries such as USA and China ESRI (2008). However, there may arise problems when using the representation. The most important problem is that it is not sufficiently simplified for the purposes mentioned above, since many details that are unimportant can be ignored. For example, links that are not directly connected with the important roads can be removed. Many entry and exit links at a junction or a choice point make it difficult to specify the number of connections between entry links and exit links.

Due to the problems with the common representation of road networks, a new road network representation by using so-called *mainlinks* and *accessorlinks* illustrated in figure 3.2, will be presented in this section. The new network representation consists of a set of junctions, and each one has mainlinks and accessorlinks. Each mainlink and each accessorlink only belongs to a single junction. In figure 3.2, it is shown that a junction consists of twelve incoming accessorlinks and four outgoing mainlinks.

There are many advantages of using the new approach to represent road networks. First, it allows one to control at two levels. We divided a network into two levels Vrancken & Soares (2007): Junctions and Links. The network consists of a set of junctions which are connected by road segments. For the local traffic control, the control of traffic within the junction is for a large part independent of control of traffic on the rest of the network. Network level traffic control is highly needed as local traffic

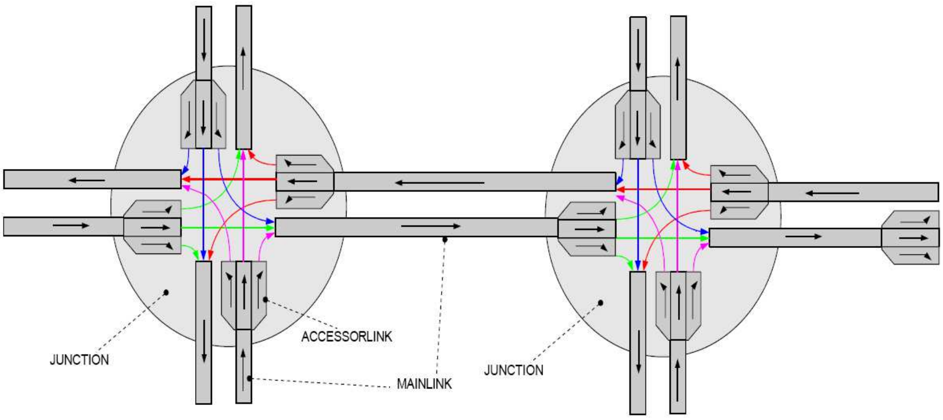


Figure 3.2: Mainlinks and accessorlinks.

optimization often causes traffic problems in other nearby location. The network level control can be done at the higher level – control coordination of junctions.

Second, the number of accessorlinks represents the number of choices at the choice point. Thus, it is possible to express the traffic data on the accessorlinks for different direction choices such as turning rates, capacity and density. These traffic data are useful for traffic control, such as traffic prediction and traffic light planning.

Third, accessorlinks and mainlinks are connected directly, thus they directly pass traffic data with each other which make the information transmission faster than the common approach, since traffic data are passed between links via nodes for the common approach. By using the mainlinks and accessorlinks structure, the amount of data flow in traffic control software is also considerably reduced. In general, it improves the performance of the traffic control system.

Last, there is only one accessorlink between two mainlinks, thus it is an 1-to-n relationship for mainlinks and accessorlinks. It means that one mainlink is connected with several accessorlinks. The other way around, it is an n-to-1 relationship for accessorlinks and mainlinks. That means several accessorlinks are connected with only one mainlink. It avoids the n-to-n relationship and makes software data structures much simpler.

With these advantages, the new road network representation has many advantages compared with other approaches. In order to reduce the work load and to prevent errors, a software generator was developed to automatically generate the road network representation.

3.3 Road network representation software generator

In this section, a software generator which was implemented in C++ will be explained. The system overview, the class diagram and the design procedure of the software generator will be explained in the following subsections. Finally, the results of the software generator will be shown.

3.3.1 System overview

The system overview of the software generator is shown in figure 3.3. The software generator reads data from the detailed Dutch road network database *Nationaal Wegenbestand*(NWB) NWB (2002). The main file (.shp) contains the primary geographic reference data in the shapefile ESRI (2007). More precisely, it contains coordinates of multiple points for each road segment. The .shx contains a positional index of the feature geometry to allow seeking forwards and backwards quickly. The .dbf file contains attributes for each road segment. The most important attributes are WVK_ID (ID of segment), JTE_ID_BEG (ID of begin node), JTE_END (ID of end node), WEGNUMMER (road number), BAANSUBSRT (type of road), RPE_CODE, ADMRICHTNG, RIJRICHTNG (these three attributes determine the road direction), STT_NAAM (street name), BEGINKM (at the begin node the absolute kilometer value), EINDKM (at the end node the absolute kilometer value) and Length (length of segment in meter). The NWB data is the ideal data source to create road network representation due to the fact that all the necessary information of road segments are contained in the database. Based on all these information in the NWB data, the software generator converts the data into the link styles which we mentioned in the previous section and stores the data in a MySQL database.

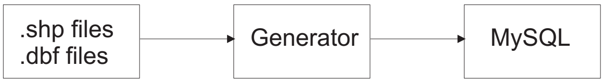


Figure 3.3: System overview.

3.3.2 Class diagram

Five main classes are defined in the software generator: Point, Node, Link, Polyline and Junction. The class diagram is shown in figure 3.4. The attributes of the class point are the x and y coordinates. A node is represented by a point. Both links and polylines have more than one point. A link has one of two types: mainlink and accessorlink. One junction is represented by the center point of the junction. Polylines contain the multiple points of links and junctions which can be used to create the images of links and junctions. From the above descriptions, we can see that the attributes of the classes contain all the information from the NWB data and all the information for the new network representation. Thus, all data for both the NWB and the MySQL database can be represented by a list of nodes, a list of links, a list of polylines and a list of junctions.

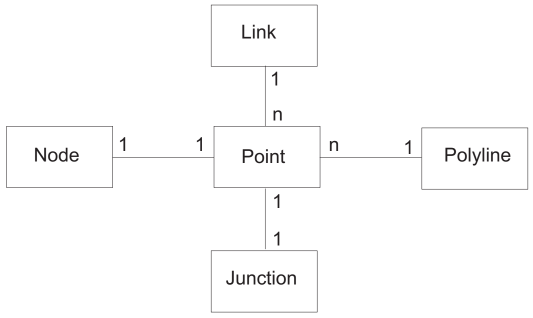


Figure 3.4: Class diagram.

3.3.3 Design procedure

Due to the fact that road segments in the NWB data might have two directions for one segment, we have to convert them into segments with only one direction for the first step. The conversion can be done based on three attributes of the NWB data: RPE_CODE, ADMRICHTNG and RIJRICHTNG. Based on the attributes, we can specify that the road has either two directions or one direction. After the link list with only one direction is created, we can assign LinkInList and LinkOutList to begin nodes and end nodes of the links. This can be done by adding link to LinkOutList of the begin node of the link and adding link to LinkInList of the end node of the link. Then we can determine TriNodes by the number of links in LinkInList and LinkOutList. TriNodes are begin nodes or end nodes of links with more than one links in LinkInList or LinkOutList. After that, we can create TriLinks. TriLinks are links starting from one TriNode and join links together until the end node of a link is also a TriNode. Finally, we can create junctions, mainlinks and accessorlinks which are the most important and also the most difficult steps.

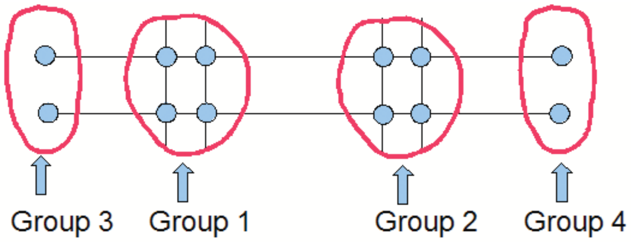


Figure 3.5: Grouped by fixed distance.

For urban roads, the diameter of junctions is relatively small (normally less than 50 meters). Normally, nodes in which the distance is smaller than a fixed length (for example 50 meters) are grouped. As it can be seen in figure 3.5, four groups are created based on the fixed distance algorithm and each group is a junction. Based on the junctions, the mainlinks and accessorlinks of the junctions are easily determined. Mainlinks are the links which connect two junctions and accessorlinks are the links which connected two mainlinks.

However, motorway junctions are larger and more complex. One example is the motorway A2-A9 crossing, in the Netherlands (Fig. 3.6). The red points on the figure are nodes which are actually beginning or end of street segments. The size of these junctions is relatively big (more than 1000 meters). If we group the nodes by tuning the parameter distance, we might have problems with finding correct nodes groups such as two junctions in one nodes group or one junction in two node groups. Another problem is that it is difficult to determine accessorlinks automatically, since in this case each accessorlink consists of more than one segment and these segments are not yet determined for each accessorlink. Thus, we cannot simply use the fixed distance based algorithm to generate the new road network representation for motorways. In this case, Dijkstra's shortest path algorithm Cormen et al. (2001) is useful to generate the network representation automatically. The detailed procedures of the network representation generation are described below.

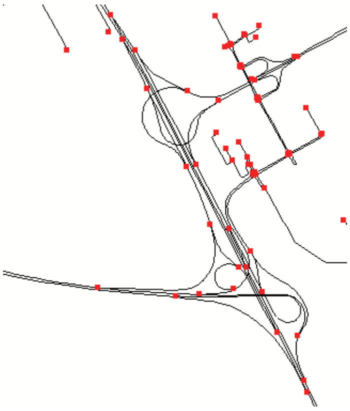


Figure 3.6: A2-A9 motorways crossing.

In order to create the junctions, mainlinks and accessorlinks of the network representation for motorway, two steps were applied. The first step was selecting nodes for the node group manually. Then, each node in the node group is connected to create the shape of the junction. The second step was to create accessorlinks for the junction by applying Dijkstra's shortest path algorithm. As normally there is only one possible path from an incoming mainlink to an outgoing mainlink in a motorway, it is possible to simply create accessorlinks by finding one possible path. Thus, Dijkstra's shortest path algorithm is useful to create an accessorlink by finding the shortest path from the incoming mainlink to the outgoing mainlink. To create all accessorlinks for the junction, the shortest paths from all incoming mainlinks to outgoing mainlinks can be found by applying Dijkstra's algorithm to the nodes. Each shortest path is an accessorlink of the junction.

Finally, all attributes of the junctions, the mainlinks and the accessorlinks are stored in a MySQL database.

The main design procedure is shown in figure 3.7. The result of the software generator is described in the next subsection.

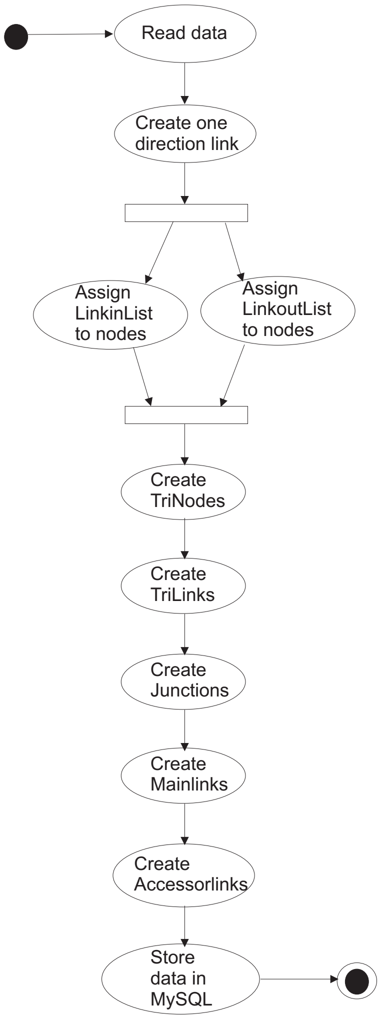


Figure 3.7: Design procedure.

3.3.4 Results

The result of the first step is shown in Fig. 3.8. The filled grey area is the shape of the junction, which was created correctly without interference by neighbouring junctions. The red links are the mainlinks which are connected with two junctions. The result of the second step is shown in Fig. 3.9. The blue links are the accessorlinks that are connected with mainlinks. All accessorlinks of the junctions were created by applying Dijkstra's algorithm. As a result, the system generated automatically all mainlinks and accessorlinks for the road network representation.

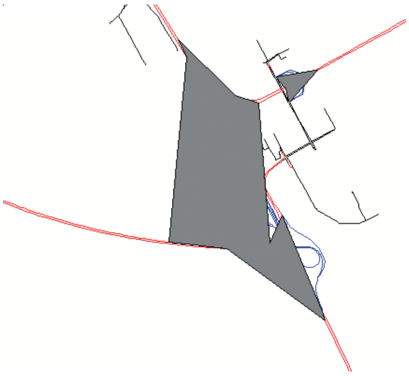


Figure 3.8: The A2-A9 after applying the first step.

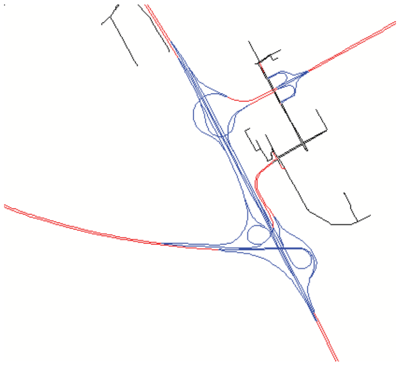


Figure 3.9: The A2-A9 after applying the second step.

3.3.5 Visualization of the new network representation

Based on the data in the MySQL database, link objects can be created and they can be shown on a geographic map automatically. We can use the motorway network around Amsterdam area as an example. The visualization of the new road network representation is shown in figure 3.10. The background of the representation is a geographic map which is supplied by a commercial map maker company. The blue links are the road network visualization which are automatically placed on the top of the geographic map based on the GPS code of the links.

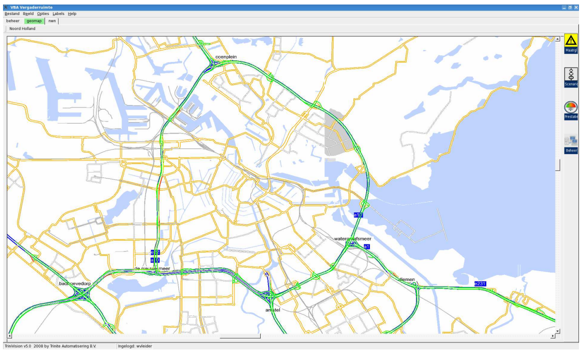


Figure 3.10: New road network representation of Amsterdam motorway network.

In summary, we created mainlinks and accessorlinks for the new road network representation based on the NWB data to reduce the complexity of road networks. In order to reduce the work load and prevent errors, we developed software to generate the network representation automatically. The network representation will be used as a base layer in a distributed architecture which will be described in the next section.

3.4 A distributed architecture for monitoring, simulation and control

A distributed architecture improves scalability, facilitating the system to grow. The logical architecture shows which elements cooperate with each other in a high level manner, without concerns about how this interaction is done. In this section, two different distributed architectures are described which will be used in this thesis.

3.4.1 The distributed architecture in the HARS project

The proposed architecture in the HARS project (see section 1.1.2) is distributed. There are several advantages in using this architecture style, such as flexibility and improved communication Tanenbaum & Steen (2002). In the HARS project, the nature of the system is inherently distributed, as links, the principal elements are distributed in the network. This is a good reason for the control system to be distributed. Thus, the design of a distributed architecture fulfills the basic requirements of the system. Also, the traffic measures are distributed with sensors and actuators spread over various locations. This allows better management in case of failures.

The network elements of the architecture are:

- **The Origin-Destination Manager** (ODMGR) represents an origin-destination pair and comprises a set of routes.
- **Routes** present a number of routes and each route consists of a sequence of links.

- **Links** a segment of road in one direction without crossings or other choice or merge points except at the begin and end point. There are two types: mainlinks and accessorlinks (Fig. 3.2).

- **Junctions** comprises the outgoing mainlinks and the incoming accessorlinks of a crossing or motorway junction. (Fig. 3.2). A junction is a location where traffic can change its routes, directions and even the mode of travel.

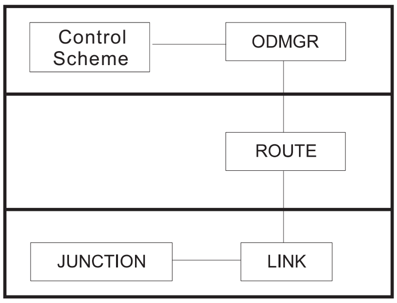


Figure 3.11: System architecture of the HARS project.

The representation of the relationship between control elements of the architecture view is shown in figure 3.11. Each road network element has a direct software object representation. The distributed components have to communicate with each other, as they work in cooperation. For instance, links have to communicate with other links in order to achieve a traffic state. They continuously measure the traffic state and communicate about it to other links in real-time.

3.4.2 The distributed architecture in the FileProof project

The proposed architecture in the FileProof project (see section 1.1.2) is also distributed (see figure 3.12). The detailed working principle between different components will be discussed in chapter 8. This chapter only explains the individual components. The high level system architecture of figure 3.12 shows the structure of the system, in which the software entities represent road network elements. These network elements are:

- **Networks**: represents a road network and consists of several subnetworks.

- **Subnetworks**: represents a subset of a road network and consists of several buildingblocks.

- **Buildingblocks**: consists of a string of connected links. It is a stretch of road in one driving direction that may contain crossings or other merge or choice points.

- **Links**: a segment of road in one direction without crossings or other choice or merge points except the begin and end point. There are two types, mainlinks and accessorlinks.

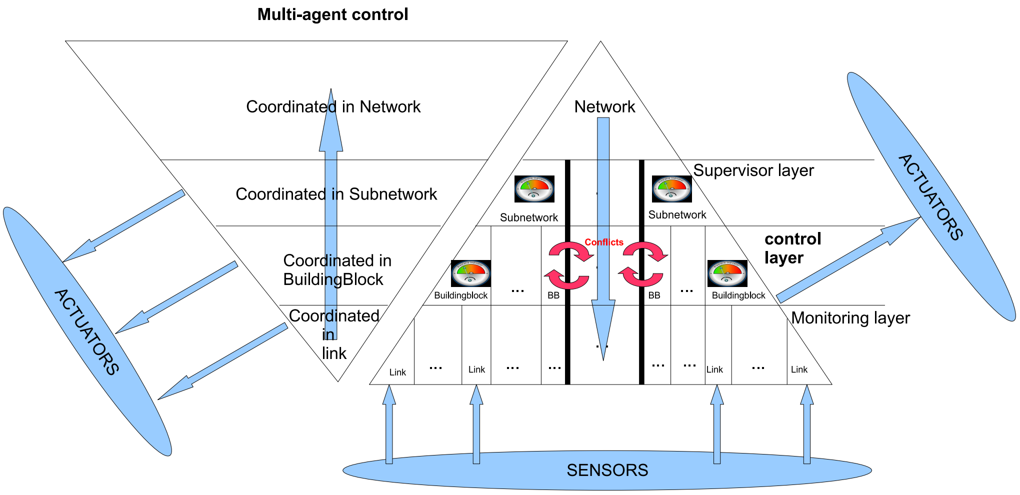


Figure 3.12: System architecture of the FileProof project.

- **Sections**: links are further divided into sections of a length of about 100 to 500 m. for the purpose of traffic prediction and control.

As we can see from figure 3.12, there are three different layers in the system: the monitoring layer, the control layer and the supervisory layer. In the monitoring layer, links receive the traffic data from sensors and filter the data using data filtering techniques Treiber & Helbing (2002). In the control layer, buildingblocks determine their states based on the traffic data of links and control different kinds of actuators (traffic control measures). In the supervisor layer, operators can monitor the state of subnetworks and operate with a selected scenario by the system. In the multi-agent control layer, links communicate with the neighboring links and coordinate with each other by using so-called *Service*. If the links are not able to handle the traffic, buildingblocks will take over and coordinate with the neighboring buildingblocks by using *Service*. If buildingblocks are still not able to handle the traffic, subnetwork will take over with a different scenario and coordinate with the neighboring subnetwork to resolve the conflicts.

In this section, two types of distributed system architectures are presented. Both system architectures can co-exist in a software application.

3.5 Distributed publish-subscribe middleware

The distributed software objects which are described above must communicate with each other by asking and providing services. There are many different means of implementing process communications Tanenbaum & Steen (2002). The most efficient communication mechanism is the Publish-Subscribe type Fiege et al. (2006); Eisenhauer et al. (2006).

Dynamic Subscribe System (DSS) datapool Soares (2010) is the publish-subscribe middleware, developed by Trinité automation. It provides a level of abstraction, by hiding the complexity of a variety of platforms, networks and low-level process communication. Application developers may concentrate on the current requirements of the software to be developed, and use lower-level services provided by the middleware when necessary. Objects may subscribe for information, unsubscribe, publish information, and notify that they are interested in some kind of information. All of these events are handled by a slot. The publish-subscribe communication mechanism will support an asynchronous (non-blocking), many-to-many communication between components in the network.

The implementation architecture is displayed in Figure 3.13. Trinivision is the graphical user interface which is presenting DSS visual objects. It can be used in both personal computers and smart phones. Each real world, physical entity has a corresponding software component named ”bridge”. Bridges are used as communication links between the environment and the system. For example, a MySQL bridge is the interface between the DSS Datapool and MySQL database. DSS uses it to read/write data from/to MySQL database.

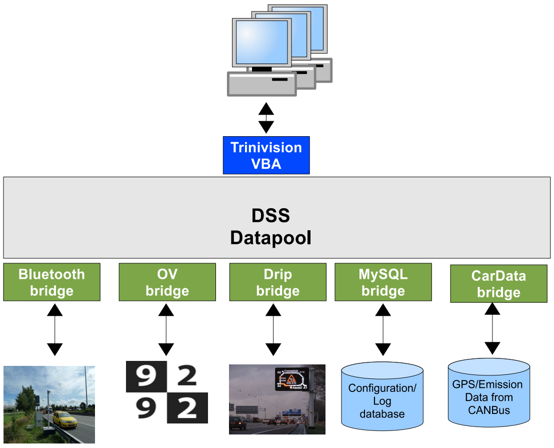


Figure 3.13: System archiecture.

Besides the middleware functionality the DSS datapool also addresses the business logic. All designed objects are implemented in the datapool. Figure 3.14 zoomed in into a small part of the DSS datapool. The balloons represent the objects, the lines represent the relation between each other. The objects within the DSS datapool have been explained above.

3.6 Conclusions

In this chapter, a new road network representation is presented to reduce the complexity of road networks. Then a software generator is explained to generate the new

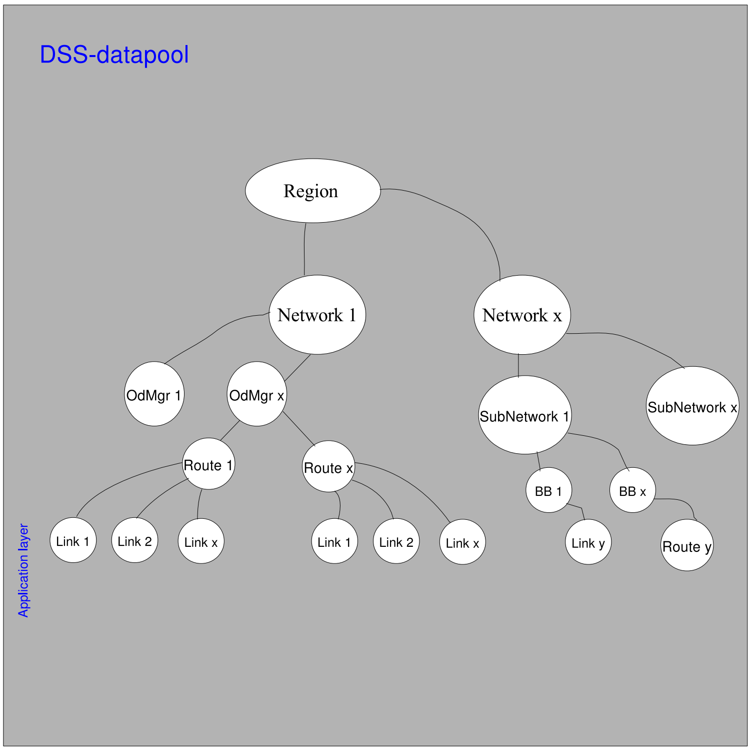


Figure 3.14: Objects relations in DSS.

road network representation for both urban road and motorways automatically, in order to reduce the work load and prevent errors. Next, two types of distributed system architectures with multi-layers based on the presentation are described. Both system architectures can co-exist in a software application. In the end, a distributed publish-subscribe middelware called DSS datapool is introduced to help developers to focus on the implementation of the current requirements by hiding the complexity of low-level process communication. The road network representation with the distributed architectures and middleware will be used for three purposes in later chapters: traffic monitoring, traffic simulation and traffic control.

Chapter 4

Traffic Flow Prediction at the Boundary of a Motorway Network[1]

4.1 Introduction

In the Netherlands there are five traffic control centers for on-line monitoring and control of motorway networks. To assist the operators with their control tasks Wang et al. (2010b, 2009a), predictions of traffic flow have to be generated for a time period. For this investigation, we think of a period in the order of 30 minutes.

The usual approach of traffic forecasting is a network-based model that simulates the traffic based on spatial information, temporal information and traffic flow dynamics Smith et al. (2002). There are many different types of network-based models in the literature such as microscopic, mesoscopic and macroscopic traffic models Smulders (1986); Messmer & Papageorgiou (1990); Lighthill & Whitham (1955); van Lint et al. (2008); Wang et al. (2010a). The network-based prediction model can be used as a decision support system to evaluate the impact of different control scenarios for the purpose of both off-line scenario planning and on-line scenario selection.

However, every network must have boundary points which do not have knowledge of the traffic state upstream from those boundary points. Thus, the boundary points cannot use spatial information and traffic flow dynamics of upstream points to predict their state. In fact, a network-based prediction model must have the information of the traffic demand at these boundary points for a future horizon.

One of the challenges of traffic demand prediction on a boundary point is to develop a simple effective algorithm which can be easily implemented in large-scale, real-time environments. The sensitivity test of the demand variation in Corthout et al. (2011) shows that 50 veh/h demand variation leads, for 2.9% of the links, to a corresponding change in the speed of 5 veh/h and in 11.9% of the links to a change in the speed of

[1] This chapter is based on my publications Wang et al. (2011a, 2014a)

more than 10%. For this reason, the accuracy of traffic demand prediction at boundary points plays a key role in supporting a network-based prediction model. Moreover, fast computation is very important for on-line large-scale traffic prediction. Thus, accurate and fast traffic demand prediction on boundary points is crucial to an effective and proactive traffic control system, more specifically, for a decision support system.

In this chapter, the design and the performance of an adaptive prediction algorithm is presented for the purpose of short-term traffic demand forecasting at a boundary point of a large-scale motorway network. Boundary points can also be exit points and the same algorithm can be applied at exit points. However it is out of scope of this chapter.

This chapter is organized as follows. Section 4.2 contains a problem description. Section 4.3 gives a literature review about the existing approaches and describes the proposed approach in this chapter. Section 4.4 describes the adaptive prediction algorithm for motorway traffic flow. Section 4.5 describes the choices of the parameters of the algorithm and illustrates the performance of the algorithm on real traffic data. Concluding remarks are stated in Section 4.6.

4.2 Problem statement

The purpose of this section is to describe in detail the adaptive prediction problem of motorway traffic flow solved in the chapter.

This research was inspired by Trinité Automation B.V. company who develops software for traffic control centers in The Netherlands, such as the center in the province of North-Holland. The road network of this traffic control center consists of the ring road of Amsterdam, the motorways A8 and A7 from Amsterdam to the north, the motorway A9 to the northwest of Amsterdam to the cities of Haarlem and of Alkmaar, the motorway A4 to the southwest along the airport Schiphol of Amsterdam to the cities of The Hague and Rotterdam, the motorway A2 to Utrecht, and the motorway A1 to the east in the directions of Hilversum, Amersfoort, and Germany, and the motorway A6 to Almere, Lelystad, and Groningen. The length of the network is about 500 km. And the geographical extent is about 80 by 50 km in respectively the north-south and the east-west directions. About 2.5 to 3 million people live in the area.

The Traffic Control Center of North-Holland monitors and stores the traffic flow data on the motorways and on several provincial roads on an hourly, daily, weekly, and yearly basis. Control measures like on-ramp metering, dynamic speed control, routing advice, and other measures are taken partly automatically and partly by the road operators.

The problem treated in this chapter is then to produce predictions for a horizon of about 30 minutes of the traffic flow at the entry points of a motorway network based on online measurements of the average traffic flow in short periods, like 10 minutes. Control theory offers for this an adaptive predictor used for several other prediction problems.

A prediction of traffic flow is by definition an approximation of future values of the observed flows. In most traffic circumstances the predictions described below are good approximations, better than last weeks traffic flow data and better than the week profile of the average of the last weeks. In case of unforeseen events affecting the traffic flow, the adaptive predictions quickly adjust.

4.3 Modeling approaches

A wide range of modelling approaches has been applied to traffic demand prediction on a boundary point. In general, these approaches can be classified into the following categories: the parametric techniques include historical average Stephanedes et al. (1981), Kalman filtering Okutani & Stephanedes (1984), exponential smoothing Messer (1993), Seasonal Autoregressive Integrated Moving Average(SARIMA) model Williams (1999) ;Smith et al. (2002), Space-Time AutoRegressive Integrated Moving Average (STARIMA) Ding et al. (2011), multi-variable time-series model Williams et al. (1998), structural time-series model Harvey (1989), Durbin & Koopman (2001); Ghosh et al. (2009) and chaotic theory Hu et al. (2003); the nonparametric techniques include non-parameter regression (NPR) models Davis & Nihan (1991), fuzzy Logic System Zhang & Ye (2008), artificial Neural Network (ANN) Siegelmann & Sontag (1991), Liu et al. (2006), Support Vector Machines (SVMs) Vapnik (2000), Davarynejad et al. (2011); Hong et al. (2010); Ding et al. (2002) and Support vector regression (SVR) Hong et al. (2006).

Although the literature offers abundant traffic demand prediction algorithms, it is difficult to choose the most suitable one for a single boundary point of a motorway network in large-scale, real-time environments. The three main reasons are as follows:

- The different road types. One algorithm that performs well for urban roads might function degradedly or even fail for motorways;
- The widely varying traffic conditions. One algorithm that performs better than another algorithm under a specific traffic condition might function degradedly or fail under other traffic conditions (such as the weather);
- The trade-off between accuracy and computation speed. An algorithm that has a high accuracy may take a long computation time.

The existing approaches have their advantages under certain circumstance. A comparison of different approaches has been discussed in the literature. For example, a comparison of seasonal autoregressive integrated moving average (SARIMA) and nonparametric models for traffic flow forecasting is available at Smith et al. (2002) with the conclusion that the performance of SARIMA models is superior to that of nonparametric regression. In Davarynejad et al. (2011), two different approaches are compared to predict short-term traffic demand. The results show that the prediction accuracy of SVM is slightly better than that of an adaptive filter.

However, the existing approaches have their own drawbacks. For example, the historical average approach has a key drawback in responding to unexpected incidents. The Kalman filtering approach is prone to produce overshoots that reduce prediction accuracies when the traffic state changes dramatically. The ARIMA approach needs to search for best orders of autoregressive and best fits of parameters in the model. The time-series models based on state-space methodology have a similar drawback as the ARIMA approach. It is difficult to achieve accurate prediction for all kinds of traffic conditions due to the fixed parameters in the models. The performance of most of the nonparametric techniques such as ANN highly depends on the amount and quality of training data. Moreover, the models are computationally demanding which means relatively long computation time.

In order to achieve accurate and robust traffic demand prediction with easy to implement and fast computation in practice for large-scale, real-time environments, we propose an adaptive prediction algorithm which is one of the most efficient solutions to this problem. There are publications on predictions of other signals, such as prediction of energy demand by customers at the level of a region or country, of prediction of gas demand in a region, or of prediction of water levels at the mouth of the Eastern Scheld in The Netherlands. The adaptive prediction algorithm for electric energy demand proposed by T. Bohlin, see Bohlin (1976), will be used in this chapter. That algorithm has also been used for adaptive prediction of railway energy demand, see Bij & Schuppen (1983). The algorithm proposed in this chapter is based on the Kalman filter and its associated predictor, but adjusted for the particular problem of prediction of traffic flow.

The approach of this chapter is as follows. For each measurement location on a motorway which gives access to the ring road of Amsterdam, measurements of the traffic flow are available for the investigation. The data are one-minute totals of the flow, the sum over two or more lanes at the selected locations. The data are aggregated into 10 minute totals. The model consists first of a week pattern consisting of seven days, 24 hours per day, and 10 minute periods per hour. After arrival of a measurement of each new 10 minute period, a prediction is made of the corresponding period one week ahead. The difference between the measurements per period and the prediction computed a week earlier are scaled by the prediction of a week earlier, and these differences are called the relative period errors. These errors are modelled as the output of a Gaussian system thus are assumed to have a memory, for which a state-space dimension of two is chosen. Based on this system, a Kalman predictor can generate predictions for one, two, or more steps ahead. This prediction procedure requires knowledge of the parameters of the system. These parameters are estimated on-line using a simple least-squares algorithm. The adaptive prediction algorithm then consists of an estimation algorithm of the parameters, a prediction algorithm of the period errors, and a prediction algorithm for the measurements.

The novelty of the chapter is in the prediction of traffic flow at boundary points of a road network and, to a minor extent, in the model of the relative period errors.

The advantages of the proposed approach are: (1) *Adaptation*. The algorithm needs little tuning because of its adaptive character. (2) *Applicability*. The algorithm can be applied to almost any motorway or on-ramp of a large road network as illustrated in the chapter. (3) *Computational performance.* The computation speed is relatively fast because of the simple equations of the algorithm. Because of this advantage, the algorithm is very well suitable for on-line applications. (4) *Prediction performance*. The algorithm has been shown to be able to predict well the traffic flow at the boundary points of a road network in almost all regular traffic circumstances.

4.4 Synthesis of the prediction algorithm

The algorithm for the prediction of traffic flow is described in this section. The subsequent sections describe the performance of the prediction algorithm.

The prediction algorithm is based on stochastic control theory, Kalman filtering, and system identification. First the Kalman filter for prediction of a Gaussian stochastic system is described. Second, the prediction algorithm is converted to that of the Kalman realization of the Gaussian system. Third, an adaptive prediction algorithm is formulated based on the Kalman predictor combined with a parameter model. Fourth, the adaptive prediction algorithm is constructed based on a least-squares parameter estimation algorithm and on the predictor algorithm. Finally the algorithm for prediction of traffic flow is described.

The formulation of the algorithm is provided below. In the next section the algorithm is tested with actual traffic flow data.

4.4.1 Prediction

In this subsection the stochastic system and the Kalman predictor are briefly described.

A *discrete-time time-invariant Gaussian system representation* is a dynamic system as understood in system theory with the representation,

$$x(t+1) = Ax(t)+Mv(t),\ x(t_0)=x_0, \tag{4.1}$$

$$y(t) = Cx(t)+Nv(t). \tag{4.2}$$

In the above representation the noise process v is a Gaussian white noise process defined to be a sequence of independent random variables each of which has a Gaussian distribution with mean value zero and variance Q_v, denoted by $v(t) \in G(0,Q_v)$. The state process takes values in $x(t) \in \mathbb{R}^n$ for n a natural number and $y(t) \in \mathbb{R}$. In this chapter, $y(t)$ is an one-dimensional variable. Gaussian systems are described in Faurre (1976); Kumar & Varaiya (1986). The identity matrix of size $n \in \mathbb{Z}_+$ is denoted by I_n and is defined as having the real number 1 at all of its diagonal elements and zeroes elsewhere.

Of a square matrix $A \in \mathbb{R}^{n\times n}$ one defines its eigenvalues as the roots of the polynomial in the variable s, $\det(sI_n - A) = 0$. The set of eigenvalues is called the *spectrum* and denoted by $\text{spec}(A)$ for a square matrix $A \in \mathbb{R}^{n\times n}$. The eigenvalues are in general complex numbers though in a special case they could be all real numbers. The matrix is called *stable* if all eigenvalues are strictly inside the unit circle which unit circle is denoted by $\mathbb{D}_o = \{c \in \mathbb{C} | |c| < 1\}$.

From stochastic realization theory it is known that a Gaussian system of the form (4.1,4.2) and its Kalman realization are equivalent, in the sense that both represent the same output process. See the references Faurre (1976); Lindquist & Picci (1985). The Kalman realization is directly related to the Kalman filter for the stochastic system (4.1,4.2). In the setting of the Kalman filter, the state $\hat{x}$ is the estimate of the state x of the stochastic system (4.1,4.2) while in a stochastic realization it is only the state of the considered stochastic realization. The fact that the stochastic realization represents also a Kalman filter will not be used in the remainder of the chapter. The reader best thinks of the system representation (4.3,4.4) as another representation of the observed process y. Therefore it seems useful to restrict attention to the Kalman realization described by the equations

$$
\begin{aligned}
\hat{x}(t+1|t) &= A\hat{x}(t|t-1) + K\overline{v}(t),\ \hat{x}(0|-1) = \hat{x}_0, && (4.3)\\
y(t) &= C\hat{x}(t|t-1) + \overline{v}(t), && (4.4)\\
& \overline{v} : \Omega \times T \to \mathbb{R}^p,\ v(t) \in G(0,\overline{q}),\ \overline{q} \in (0,\infty), \\
& x(t) \in \mathbb{R}^n,\ y(t) \in \mathbb{R}^p,\ p = 1, && (4.5)\\
& A \in \mathbb{R}^{n\times n},\ K \in \mathbb{R}^{n\times p},\ C \in \mathbb{R}^{p\times n}, \\
& \text{spec}(A) \subset \mathbb{D}_o,\ \text{spec}(A-KC) \subset \mathbb{D}_o, && (4.6)\\
\mathbb{D}_o &= \{c \in \mathbb{C} | |c| < 1\}.
\end{aligned}
$$

In the above representation $\overline{v}$ is a Gaussian white noise process with variance $\overline{q}$. In the context of a Kalman filter it is called the innovations process. Conditions (4.6) guarantee that the system is strictly stable and that the inverse system is strictly stable. The matrix K is called the *Kalman gain matrix* in the Kalman filter.

Consider a time-invariant Gaussian system in the Kalman realization representation. Consider the prediction problem on the time index set $T = \{t_0, t_0+1, \ldots, t_1\}$ and the prediction horizon at time $t \in T$, $T_p(t) = \{t+1, t+2, \ldots, t+t_p\}$, for $t_p \in \mathbb{Z}_+$ denotes the length of the prediction horizon. The reader should clearly distinguish between the time index set T and the moving prediction horizon $T_p(t)$ for $t = t_0, t_0+1, \ldots, t_p$.

The *time-invariant Kalman predictor for a Kalman realization* is defined by the following procedure.

1. For each time $t \in T$ do:
   The filtering step.

$$
\begin{aligned}
\hat{x}(t+1|t) &= A\hat{x}(t|t-1) + K[y(t) - C\hat{x}(t|t-1)], && (4.7)\\
& \hat{x}(0|-1) = 0.
\end{aligned}
$$

The prediction step. For all $s = 1, \dots, t_p$ do:

$$\begin{aligned}
\hat{x}(t+s+1|t) &= A\hat{x}(t+s|t),\ \hat{x}(t+1|t), && (4.8)\\
Q(t+s+1|t) &= AQ(t+s|t)A^T + \overline{q}KK^T, && (4.9)\\
& \quad Q(t+1|t) = 0, \\
\hat{y}(t+s|t) &= C\hat{x}(t+s|t), && (4.10)\\
q_y(t+s|t) &= CQ(t+s|t)C^T + \overline{q}. && (4.11)
\end{aligned}$$

The solution to the multi-step prediction problem is given by the formulas for $s \in \{1, 2, \dots, t_p\}$, $w \in \mathbb{R}$

$$\begin{aligned}
& E[\exp(iwy(t+s))|F_t^Y] \\
&= \exp(iw\hat{y}(t+s|t) - \frac{1}{2}w^2 q_y(t+s|t)). \qquad (4.12)
\end{aligned}$$

The initial condition of the predictor $\hat{x}(t+1|t)$ is set equal to the filter estimate produced by (4.7).

4.4.2 Adaptive prediction

For prediction of a Gaussian system of which the parameter values are not known, it is necessary to work with an adaptive predictor. Adaptive prediction is explained in this subsection. The results are based on system identification of Gaussian systems, see Ljung (1987); Söderström & Stoica (1989).

The Kalman predictor has to be transformed to another representation that is better suited to estimate the parameters of the system on-line. The transformation is well known in system theory and concerns a single-output system. The one-step Kalman predictor or the Kalman filter of the Kalman realization has the equivalent representation

$$\begin{aligned}
h(t+1|t) &= L_n h(t|t-1) + M_n \hat{y}(t|t-1) + N_n y(t) && (4.13)\\
& \quad h(0|-1) = 0, \\
y(t) &= p^T h(t|t-1) + \overline{v}(t) && (4.14)\\
\hat{y}(t+1|t) &= h(t+1|t)^T p, && (4.15)\\
q(t+1|t) &= \overline{q}, && (4.16)
\end{aligned}$$

where the following definitions are used,

$$
\begin{aligned}
& h : \Omega \times T \to \mathbb{R}^{2n},\ y : \Omega \times T \to \mathbb{R}, \\
h(t+1|t) &= \begin{pmatrix} -\hat{y}(t|t-1), \dots, -\hat{y}(t-n+1|t-n), \\ y(t), \dots, y(t-n+1) \end{pmatrix}^T, \\
p &= (\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n), \\
L_n &= \left(\begin{array}{ccc|ccc} 0 & \dots & 0 & & & \\ & & \vdots & & 0 & \\ I_{n-1} & & 0 & & & \\ \hline & & & 0 & \dots & 0 \\ & 0 & & & & \vdots \\ & & & I_{n-1} & & 0 \end{array}\right) \in \mathbb{R}^{2n \times 2n}, \\
M_n &= -e_1 \in \mathbb{R}^{2n},\ N_n = e_{n+1} \in \mathbb{R}^{2n},
\end{aligned}
$$

where $e_j \in \mathbb{R}^{2n}$ denotes the j-th unit vector in the indicated vector space. The real numbers $\alpha_1, \dots, \alpha_n$ and $\beta_1, \dots, \beta_n$ are directly obtained from the matrices A, K, C using a well known transformation from a linear system with a single output to its observable-canonical-form, see (Chen, 1984, pp. 240–242) and (Sontag, 1998, pp. 290–292).

The multistep Kalman predictor has the equivalent form,

$$
\begin{aligned}
& \text{for } s = 1,\ 2, \dots, t_p, \text{ compute,} \qquad (4.17) \\
\hat{h}(t+s+1|t) &= L_n \hat{h}(t+s|t) + (M_n + N_n)\hat{y}(t+s|t), \qquad (4.18) \\
& \hat{h}(t+1|t) = h(t+1|t), \\
\hat{y}(t+s+1|t) &= \hat{h}(t+s+1|t)^T p, \qquad (4.19) \\
Q_r(t+s+1|t) &= F(p)Q_r(t+s|t)F(p)^T + \overline{q}G(p)G(p)^T, \\
& Q_r(t+1|t) = 0, \qquad (4.20) \\
q_y(t+s+1|t) &= H(p)Q_r(t+s+1|t)H(p)^T + \overline{q}, \qquad (4.21)
\end{aligned}
$$

where the notation is used,

$$
F(p) = \begin{pmatrix} -\alpha_1 + \beta_1 & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ -\alpha_{n-1} + \beta_{n-1} & 0 & \dots & 1 \\ -\alpha_n + \beta_n & 0 & \dots & 0 \end{pmatrix},\ G(p) = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix}, \tag{4.22}
$$

$$
H(p) = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}. \tag{4.23}
$$

The proof may be found in (Bij & Schuppen, 1983, Section 3.6).

The self-tuning synthesis procedure will be used for adaptive prediction. Thus the parameter vector is modelled as the state of a Gaussian system below. Consider the system representation, where p has become time dependent, $p(t)$.

$$\begin{aligned}
p(t+1) &= Bp(t)+r(t),\ p(0), && (4.24)\\
y(t) &= h(t|t-1)^T p(t)+\overline{v}(t), && (4.25)\\
& r(t)\in G(0,v_1 I_{2n}),\ \ p(0)\in G(0,v_0 I_{2n}),\\
& p(0),\ r,\ \overline{v},\ \text{independent.}
\end{aligned}$$

Where B is a diagonal matrix with $B_{ii}\in(-1,+1)$ for all $i=1,\ldots,n_p$. That system representation is based on the above introduced Gaussian system but extended by an equation for the dynamics of the parameter vector p. The process $\overline{v}$ is a Gaussian white noise process, a sequence of independent random variables each of which has a Gaussian distribution with the indicated mean and variance.

The conditional distribution of

$$\begin{aligned}
& y(t+s)\mid\{y(0),y(1),\ldots,y(t-1)\},\\
& \text{is then approximated by the characteristic function}\\
& \exp(iu\hat{y}(t+s|t)-\frac{1}{2}u^2 q(t+s|t)),\ u\in\mathbb{R},\ \text{where,} && (4.26)\\
k_p(t) &= BQ_p(t|t-1)h(t|t-1)\times\\
& \times[h(t|t-1))^T Q_p(t|t-1)h(t|t-1)+\overline{q}]^{-1}, && (4.27)\\
& \hat{p}(t+1|t)\\
&= B\hat{p}(t|t-1)+k_p(t)[y(t)-h(t|t-1)^T\hat{p}(t|t-1)],\\
& \hat{p}(0|-1)=0, && (4.28)\\
& Q_p(t+1|t)\\
&= BQ_p(t|t-1)B^T+v_1 I_{2n}-BQ_p(t|t-1)h(t|t-1)\times\\
& \times[h(t|t-1))^T Q_p(t|t-1)h(t|t-1)+\overline{q}]^{-1}\times\\
& \times h(t|t-1)^T Q_p(t|t-1)B^T, && (4.29)\\
& Q_p(0|-1)=v_0 I_{2n},
\end{aligned}$$

and the equations (4.18,4.19,4.20,4.21) in which p is replaced by $\hat{p}(t+1|t)$. The proof follows from the conditional Kalman filter, see Chen et al. (1989).

There is a stability issue with the recursive least-squares parameter estimation algorithm. The parameter estimates will, in rather particular circumstances, diverge resulting in very large prediction errors. The author of this thesis suggests to insert a test on the innovation process to check whether or not instability takes place. If an instability is suspected then a computation will result in a new parameter vector which represents a strictly stable system with a strictly stable inverse system, hence the conditions (4.6) are satisfied. When the algorithm is properly initialized then instability does not seem to occur for the data used in this investigation.

4.4.3 Adaptive prediction of motorway traffic flow

For the prediction of traffic flow at the boundary of a motorway network an adaptive prediction algorithm will next be described.

The problem is to predict the traffic flow at a pre-specified entry point of a motorway network. Note that for a stretch of motorway, models of the dynamics are available based on hydrodynamic flow. But for the dynamics of the flow at the boundary of a motorway network there is no model available. Therefore an adaptive prediction algorithm will be proposed. Traffic flow is a positive value, we apply the approach sketched before consisting of Kalman filter which is reasonable when the trajectory is away from zero. On the other hand, it is not really interesting to have a prediction if traffic flow is close to zero.

The overall algorithm for prediction consists of the following steps: (1) a week profile and its predictor; (2) the differences between the measurements of a period within a week and the prediction of the same period computed a week earlier, the relative period errors; a model for the relative period errors in the form of a Gaussian system and an adaptive prediction algorithm for the relative period errors; (3) the prediction algorithm based on the predictions of the week profile and the predictions of the relative period errors. The full algorithm is described below.

For the model and the prediction a period is selected of 10 minutes. In the design stage the selection of the period is discussed and alternatives are considered. The model consists of a week profile with each element of the week vector representing the total inflow during a period of 10 minutes. The choice of a week profile allows for changes from week to week based on the dynamics of the traffic flow over the annual seasons. The week vector thus has dimension $n_w = 1008$ (= 6 * 24 * 7). The dynamics of the week vector is a simple system with a system matrix which is the identity matrix and a Gaussian white noise with a diagonal variance matrix.

(1) The predictor for week $s \in \mathbb{Z}$ and period $t \in T = \{1, 2, \ldots, n_w\}$ is then,

$$\begin{aligned} k_w(s) &= q_w(s|s-1)[q_w(s|s-1)+v_2]^{-1}, && (4.30)\\ &\hat{w}(s+1|s, t \bmod n_w) && (4.31)\\ &= \hat{w}(s|s-1, t \bmod n_w) + \\ &\quad +k_w(s)[y(s,t) - \hat{w}(s|s-1, t \bmod n_w), \\ &\quad \hat{w}(0|-1) = m_{w_0}, \\ q_w(s+1|s) &= q_w(s|s-1) + v_1 + \\ &\quad -q_w(s|s-1)^2[q_w(s|s-1)+v_2]^{-1}, && (4.32)\\ &\quad q_w(0|-1) = v_0, \\ q_{ya}(s+1|s) &= q_w(s+1|s) + v_2. && (4.33) \end{aligned}$$

Here q_{ya} denotes the adaptive prediction of the variance of the traffic flow y. Thus, after receipt of a measurement, the predictor produces an estimate of the week profile for that particular period next week. The algorithm is initialized by a week profile obtained by averaging the week profiles of several preceding weeks.

(2) Define the *relative error* of week s and of period t as

$$e(s,t) = [y(s,t) - \hat{w}(s|s-1,t)]/\hat{w}(s|s-1,t). \tag{4.34}$$

$y(s,t)$ denotes the traffic flow in week s and with period t of that week. The design choice for this form of the errors is motivated by the wide variation of the measurements during the day. The traffic flow is very low at night, has high peaks in the morning and the afternoon rush hour, and can be 50 % to 70 % of the morning peak size in the middle of the day.

The model for the relative period errors is that it is the output of a Gaussian system of dimension $n \in \mathbb{Z}$. The parameters of this system are not known because the dynamics is not known. In addition, the dynamics may change over time and be dependent on the location in the network where the measurements are taken. Hence an adaptive prediction algorithm as developed before is used for the relative period errors. The prediction horizon is 30 minutes hence three periods of 10 minutes. The adaptive prediction algorithms for the relative period errors is then that as specified in Subsection 4.4.2. That algorithm then produces the predictions in week s and at time t,

$$\begin{aligned} &\hat{e}(s,t+1), \hat{e}(s,t+2), \hat{e}(s,t+3), \\ &\forall\, t \in \{1,\ldots,n_w\},\ s \in \mathbb{Z}_+. \end{aligned}$$

(3) Finally the predictions of the traffic flow data are then described by the following expressions,

$$\begin{aligned} \hat{y}(s,t+i) &= \hat{w}(s|s-1,t+i) + \\ &\quad +\hat{w}(s|s-1,t+i)\hat{e}(s,t+i), \\ &\quad i = 1,2,3. \end{aligned} \tag{4.35}$$

The overall prediction algorithm is thus specified by the equations for the week profile (4.30,4.31,4.32,4.33), for the parameter estimate (4.27,4.28,4.29), and for the adaptive prediction (4.18, 4.19, 4.20, 4.21) in which p is replaced by $\hat{p}(t+1|t)$.

The design parameters of the algorithm are (1) the period over which the measurements are aggregated, chosen above to be 10 minutes; (2) the dimension of the Gaussian system for the relative period errors which represents the memory of the system; (3) the variances of the adaptive prediction algorithms. The values of these design parameters are determined for particular measurements in the next section.

4.5 Testing of the prediction algorithm

In the case study, the ring road of Amsterdam (Figure-4.1) is considered. In Wang et al. (2010a), a multi-class first-order traffic flow model van Lint et al. (2008) is used to simulate the traffic of the ring road in a distributed way. However, the traffic flow at the boundary of the ring road is unknown for the next 30 minutes. Thus, in Wang et al. (2010a) it was assumed that the next 30 minutes traffic flow is the same as the current measurements. Those inputs deviate considerably from later measurements. In order to predict the traffic more accurately, the traffic flow on the boundary of the ring road is needed. Adaptive filtering is used to predict the traffic flow for the next 30 minutes. The prediction algorithm consists of three steps as described in the previous section: prediction of the week profile, short term prediction of the relative errors and short term prediction of the traffic flow. The core MATLAB source code can be seen in A.1.

4.5.1 The prediction setting

There are four motorways (A1, A2, A4, and A8) which end on the ring road of Amsterdam (Figure 4.1). And there are 22 on-ramps on the ring road. Thus, there are two kinds of inflows on the boundary network: (1) from motorways; (2) from on-ramps. In the chapter, we will first focus on (1). Then we will explain the problems and the solution when the same algorithm is applied for (2).

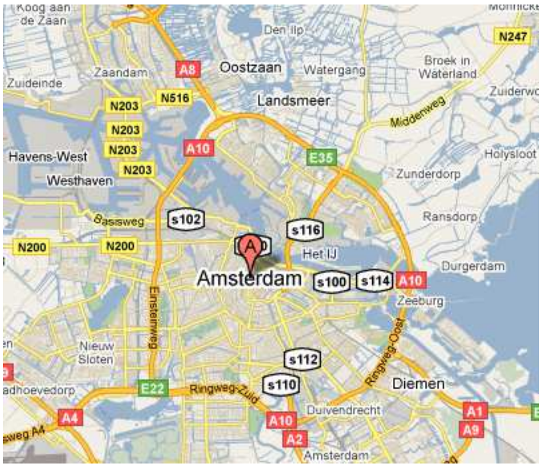


Figure 4.1: The ring road of Amsterdam.

4.5.2 The data for testing for traffic flow from motorways

In order to test the algorithm for traffic flow from motorways, we collected traffic flow data from four sites on the motorways ending on the ring road of Amsterdam a short distance before the merge point. The data on the four sites (A1, A2, A4, and A8) were collected from 20th May 2010 until 24th June 2010. The one-minute average traffic data over five weeks were collected by the MONICA sensors (velocity-flow measurement points). The data of five weeks are displayed superimposed in Figure 4.2. There are similarities in the traffic patterns of the five different weeks. The week profile has been computed using traffic data of the first four weeks in a recursive way. The fifth week of traffic data has been used for the adaptive prediction.

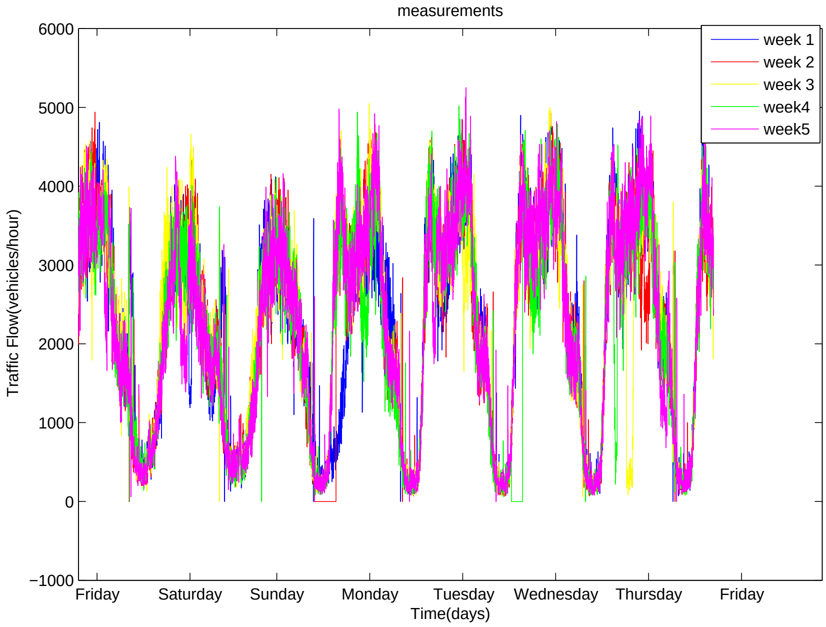


Figure 4.2: Traffic flow data with one-minute totals of five weeks superimposed.

The evaluation criteria: The evaluation criteria in terms of the computation time will be not addressed in detail, because the computation time is rather small (in the order of 0.01 second). For the evaluation of the accuracy several evaluation criteria are used. As with any system identification and prediction problem, the main evaluation criteria are the value of the *variance of the innovation process* and the difference between the *correlation function of the innovation process* and that of a white noise process. The innovation process is the difference between the measurement and the one-step prediction. Another criterion is the *variance-accounted-for* (VAF) which is defined as the ratio (100 means fully explained, 0 means no explanation) of the difference of the variance of the measurements and that of the innovation process over the variance of the measurements. Two more criteria are *Root mean square error (RMSE)* and *Root mean percentage error (RMPE)*. RMSE is a frequently-used measure of the differences between values predicted by a model or an estimator and the values actually observed from the data being modelled or estimated. RMPE is similar to RMSE but expressed as a percentage, where lower values indicate less residual variance.

The variance accounted for, RMSE, and RMPE will also be computed with respect to the week profile to establish to what extent the predictions are an improvement over the week profile.

4.5.3 Design decisions

The algorithm for the adaptive filter has been explained in the previous section. However, some design decisions are still needed. There are three main design decisions:

Decision 1: Aggregation of time average

Since the adaptive filter needs to predict the traffic flow for the next 30 minutes, it is not of interest to predict the minute-by-minute fluctuation. Five different values for the aggregation number were chosen for testing purposes: 1 minute, 5 minutes, 10 minutes, 15 minutes and 30 minutes. The traffic flow data for the first week with these numbers are displayed in Figure 4.3.

Figure 4.4 shows the profile of Tuesday's traffic flow which is zoomed in from Figure 4.3.

The reader should notice from the figure with Tuesday's traffic flow that there is still too much fluctuation for the 5 minutes aggregation. However, the fluctuation is much less for the 10 minutes aggregation. So 10 minutes total is chosen for the demand prediction. In order to predict the demand for the next 30 minutes, 3 prediction steps of 10 minutes are needed.

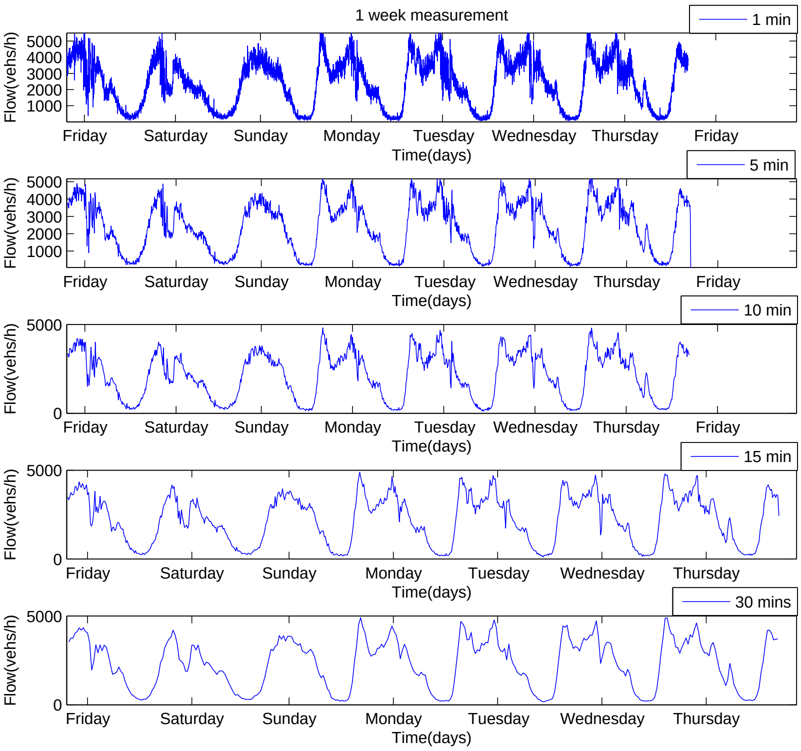


Figure 4.3: Different aggregations of a week of traffic flow.

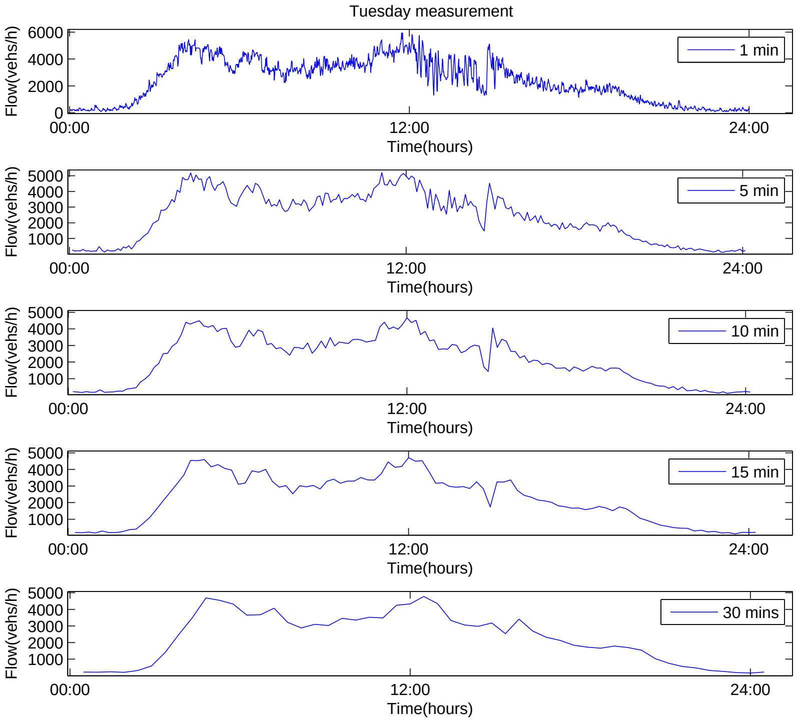


Figure 4.4: Different aggregations of Tuesday's traffic flow.

Table 4.1: The VAF value of predictions.

	1st step prediction	2nd step prediction	3rd step prediction
n=1	95.3	92.3	88.0
n=2	95.3	92.3	88.1
n=3	94.9	92.0	88.2

Decision 2: The memory of the predictor

The dimension of the state space of the predictor can be chosen. The values $n = 1$, $n = 2$, and $n = 3$ were considered. For each case, the VAF was calculated (Table 4.1). The VAF value of the week profile is 88.5. Memory $n = 2$ is chosen based on the VAF value, since memory $n = 1$ is too short and memory $n = 3$ is not needed for these data. The correlation function of the innovations is also checked, the correlation function converges quickly to zero in about two steps.

Decision 3: Variances of the noise in the Kalman predictor

Since the Kalman predictor is a single-output predictor, the noise is also of dimension one. Both the state components and the measurements are traffic flow. The stochastic system used for the parameter estimation is described by equations (21,22). The

Table 4.2: STD of different $\overline{q}$ and $v1$.

	STD of the relative error $e(t)-\hat{e}(t)$
$\overline{q}$=0.9, v_1=0.09	0.1269
$\overline{q}$=0.09, v_1=0.9	0.3198(Predictions are unstable)
$\overline{q}$=0.9, v_1=0.009	0.1256
$\overline{q}$=0.9, v_1=0.0009	0.1256

variances of the noise processes are v_1 and $\overline{q}$, see also equations (4,5). Instead of determining v_1 and $\overline{q}$ separately, it suffices to determine their ratio. Therefore a small range of values was chosen for this ratio as shown on Table 4.2.

The variance of the prediction error is also shown in table 4.2. Based on that table, the values ($\overline{q}$, $v1$)=(0.9, 0009) were chosen. The associated eigenvalues of the matrix $A-KC$ are (0.6030, -0.3134) which are both stable.

4.5.4 Performance evaluation in case of motorway traffic flow

The five weeks of traffic flow data with 10 minutes aggregation are displayed in Fig. 4.5. From the figure, we can more clearly see the similarity in the traffic pattern for the five different weeks. The peaks of the figure from left to right show the rush hours of Friday, Saturday, Sunday, Monday, Tuesday, Wednesday, Thursday and part of Friday.

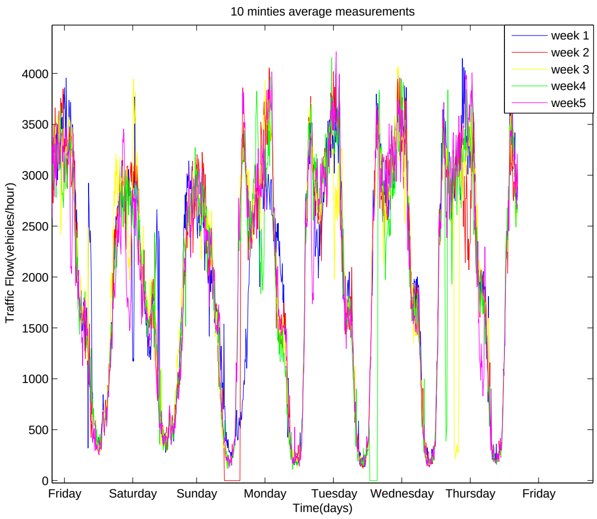


Figure 4.5: Five weeks of traffic flow data with 10 minutes aggregation.

The week profile update

The first four weeks are used to compute the profile. The week profile is compared with the last week measurement in Figure 4.6. We can see from the figure that the error between the week profile and the last week measurement is still relatively large. As we explained above, the relative period error will be used to predict the traffic flow. Figure 4.7 shows the relative error. The relative period errors are relatively small, most relative errors are between -0.2 and 0.2. The spikes in the relative errors are due to the fact that there are not too many cars on the road at night. The reader should note the memory present in the relative period errors which are taken care of in the short term prediction. Performance of the short term prediction based on the relative error will be explained in the following subsection.

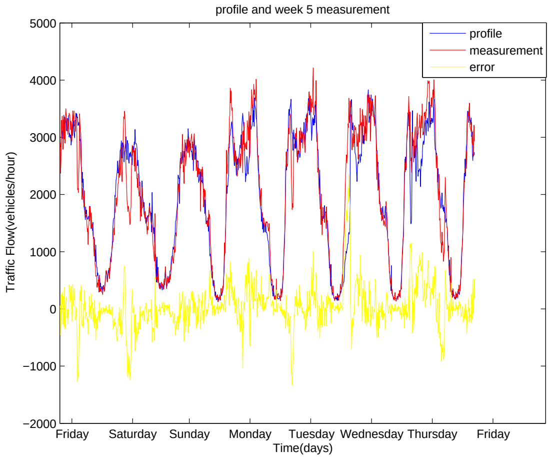


Figure 4.6: Errors between week profile and last week measurement.

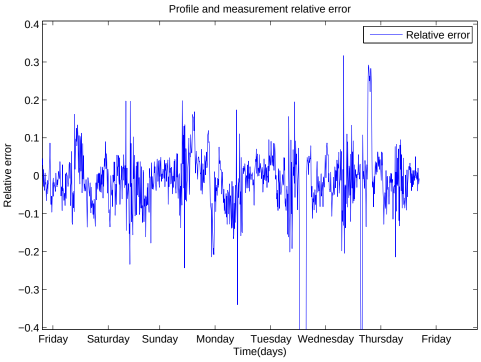


Figure 4.7: Relative errors between week profile and last week measurement.

The short term predictions of the relative errors

The relative errors are predicted based on the equations of subsection 4.4.3.

The short term predictions of the traffic flows

Finally the prediction of the traffic flow is calculated based on the equations of subsection 4.4.3. The predictions of different days can be seen in Figure 4.8, Figure 4.9, Figure 4.10, and Figure 4.11. As we can see from these figures, the traffic flow can be well predicted even in case of congestion on Tuesday and Wednesday.

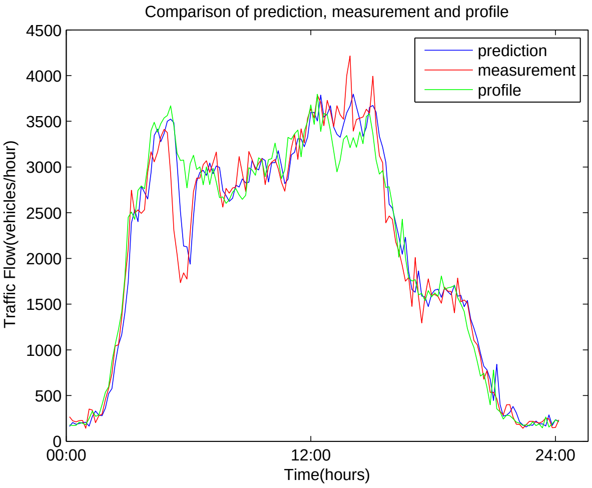


Figure 4.8: Adaptive predictions of Tuesday's traffic flow.

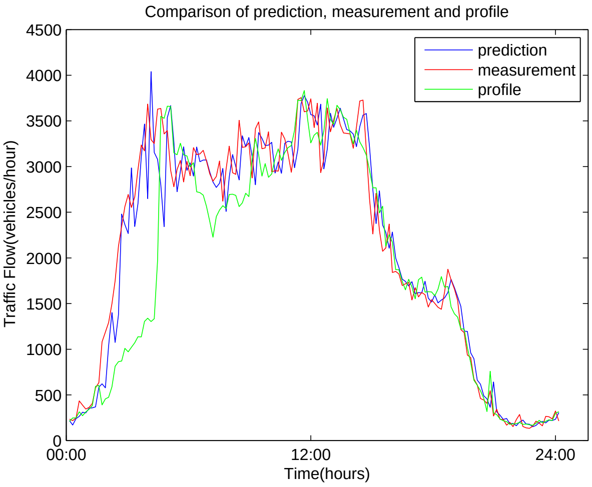


Figure 4.9: Adaptive predictions of Wednesday's traffic flow.

At each time step, one produces predictions over a horizon of 30 minutes, hence three steps. These 3-step predictions are displayed in Figure 4.12 for every time step in the period from 10:00 to 12:00 on Monday. The circles represent the week profile, the squares represent the current measurement. The . , x and * represent the first step prediction, the second step prediction and the third step prediction, respectively. The red lines are drawn based on the 3-step predictions. We can see from the figure that the 3-step predictions are much closer to the measurements than the profile.

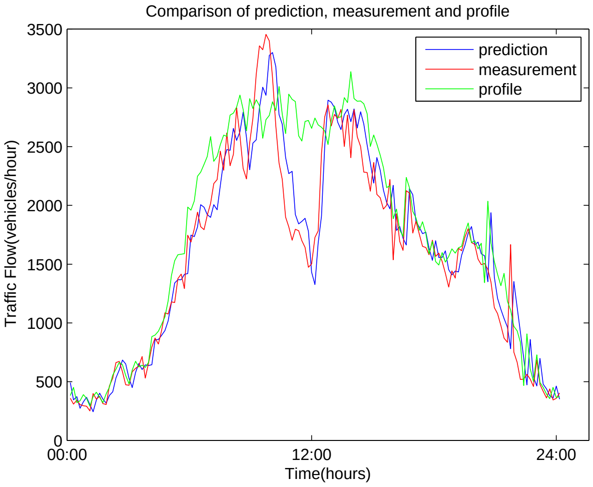


Figure 4.10: Adaptive predictions of Saturday's traffic flow.

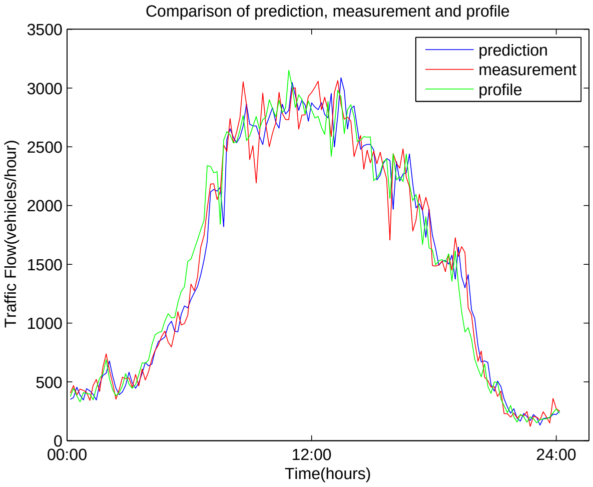


Figure 4.11: Adaptive predictions of Sunday's traffic flow.

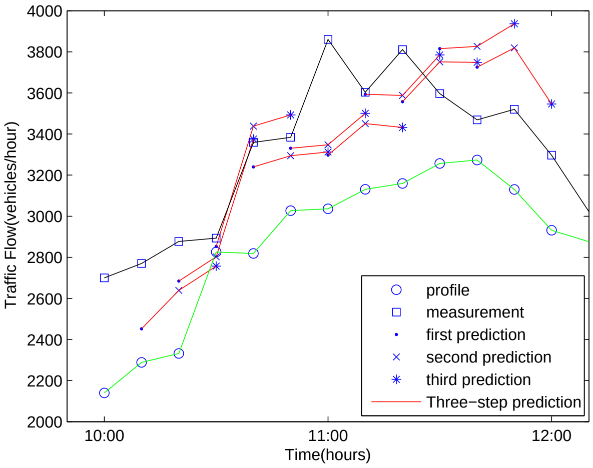


Figure 4.12: Three-Step traffic flow predictions. One step corresponds to 10 minutes.

Table 4.3: Comparison of the adaptive predictor and of the week profile.

	1st step prediciton(SP)	2nd SP	3rd SP	Week profile
RMSE	247.6	315.4	386.4	390.0
RMPE	12.4	14.2	15.3	15.1
VAF	95.3	92.3	88.1	88.5

Table 4.4: Comparison after the improvements.

	1st step prediciton(SP)	2nd SP	3rd SP	Week profile
RMSE	247.6	312.5	356.0	390.0
RMPE	12.4	14.1	15.1	15.1
VAF	95.3	92.3	90.0	88.5

4.5.5 Performance of the algorithm in case of motorway traffic flow

The results of the demand prediction by using the adaptive filter are compared with the week profile. Table 4.3 shows the RMSE, RMPE and VAF values of the adaptive filter and the week profile. We can conclude that the adaptive filter can better predict the traffic flow than the week profile for the first and the second step prediction, however the week profile is approximately the same as the third step prediction of the adaptive filter.

4.5.6 Further improvements in practical implementation

The predictions of the traffic flow at the considered locations will always take values between zero and the capacity of the road. The predictions produced by the adaptive filter may not satisfy those conditions. Therefore the adaptive prediction algorithm is modified such that negative prediction values are set to zero, while values exceeding capacity are set equal to the capacity of the road. Thus, the performance will be further improved if this constraint is added in the adaptive filter calculation. Table 4.4 shows the RMSE, RMPE and VAF values of the adaptive filter and the week profile after the constraint has been applied. We can see that there is almost no change for the first and the second step prediction, however there is a big improvement for the third step prediction. The reason is that there are many prediction overshoots in the third step prediction and they are removed by application of the constraint. Eventually, all 3-step predictions are better than the week profile. We can conclude that the adaptive filter can better predict the traffic flow than the week profile.

4.5.7 The data for testing for traffic flow from on-ramps

In order to test the algorithm for traffic flow from on-ramps, we collected the traffic flow data from four on-ramps on the east ring road of Amsterdam. Due to technical reasons, we cannot collect the data from on-ramps directly. Thus, we decided to use an alternative way to collect the data by using the difference between downstream flow and upstream flow of on-ramps. The data on the four on-ramps were collected from 14th March 2011 until 10th April 2011. The one-minute average traffic data over four weeks were collected by the MONICA sensors (velocity-flow measurement points). The data of four weeks are displayed superimposed in Figure 4.13. The data of the first three weeks have been used to update the week profile and the data of the last week have been used for the short term predictions.

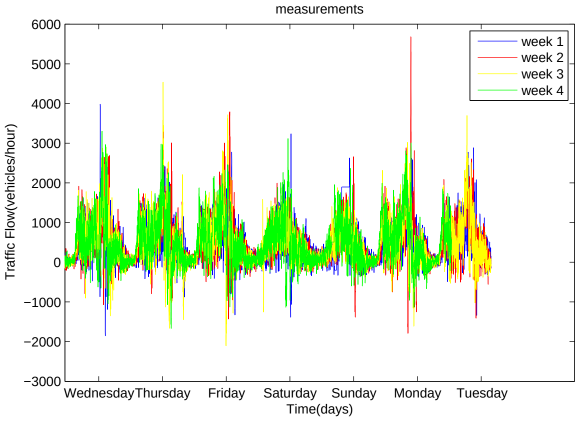


Figure 4.13: Traffic flow data with one-minute totals of four weeks superimposed.

4.5.8 Problems in applying adaptive filter for on-ramp traffic flow

Predictions are also needed for the on-ramp traffic flows. The adaptive prediction algorithm described in section 4.4.2 was also applied to traffic data of on-ramp flows. In several traffic situations the predictions are extreme large or small and hence are not directly useful. The cause of this phenomenon is primarily the relatively low value of the flows combined with the relatively high variance or fluctuations of the flows. The last factor involves irregular spikes and high-frequency disturbances. Therefore it was decided to pre-process the traffic flow data before applying the adaptive filter algorithm.

4.5.9 Solution: the pre-processing procedures

The pre-processing procedures contain two steps: remove spikes and remove high-frequency noise. The spikes are removed by setting their values to either a maximum or a minimum value. In the chapter, a low pass filter called a Butterworth filter will be used to remove high-frequency noise. We can see from the frequency response diagram from Figure 4.14 that there are high-frequency disturbances or noise in data. We can conclude that the high-frequency (higher than 0.2) data must be removed. We used the same procedure for the design decisions which we applied to the motorway data. The decisions (which are the same as for the motorway data) are as follows:

- Aggregation of time average: 10 minutes aggregation
- Memory of the predictor: $n = 2$
- Variances of the noise in the Kalman predictor: $\overline{q} = 0.9$ and $v1 = 0.009$

The comparison of the original data and the filtered data of the first week can be seen in Figure 4.15. Figure 4.16 shows the Thursday's traffic flow which is zoomed in from Figure 4.15. The figure shows that the filtered data become much more smooth than the original data. In the end, the same adaptive filter as we described before has been applied.

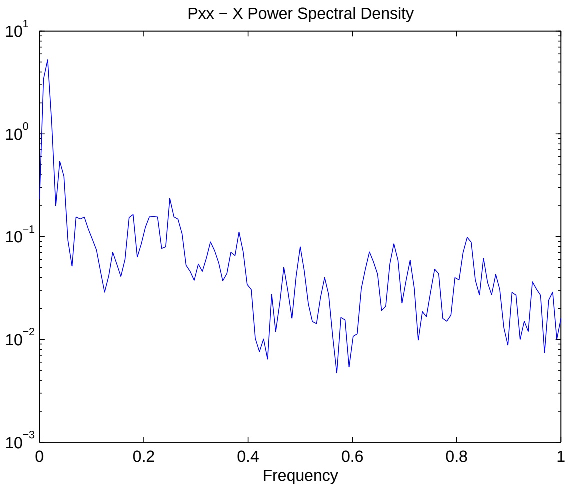


Figure 4.14: The frequency response diagram.

4.5.10 Performance evaluation in case of on-ramps traffic flow

The four weeks of traffic flow data with 10 minutes aggregation are displayed in Figure 4.17. In the figure, we can more clearly see the similarity in the traffic pattern for the four different weeks. The peaks of the figure from left side to right side shows the rush hours of Wednesday, Thursday, Friday, Saturday, Sunday, Monday, Tuesday.

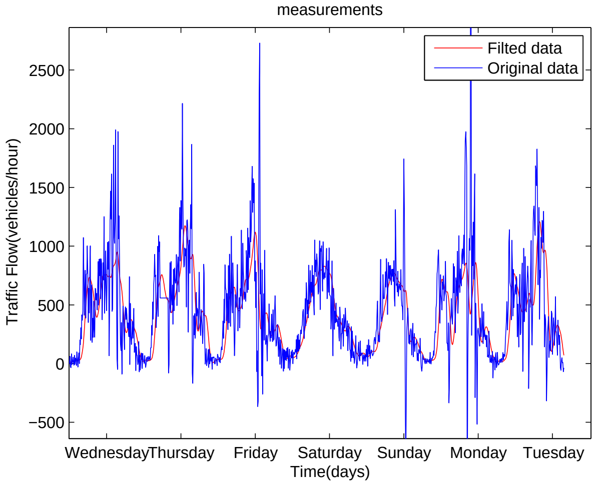


Figure 4.15: Comparison of the original data and the filtered data.

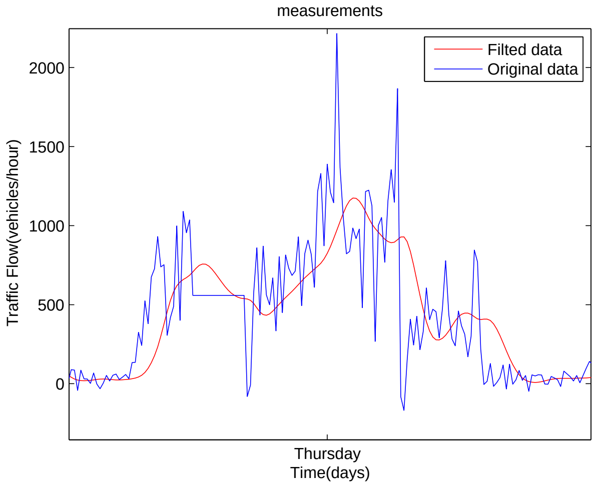


Figure 4.16: Comparison of original data and filtered data on Thursday.

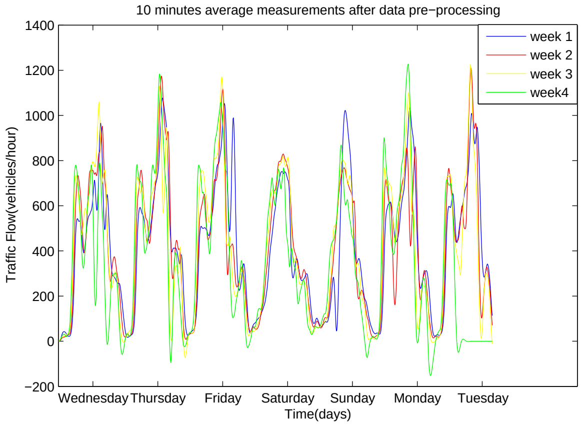


Figure 4.17: Four weeks of traffic flow data with 10 minutes aggregation.

The week profile update

The week profile has been computed using traffic data of the first three weeks in a recursive way. The fourth week of traffic data has been used for the adaptive prediction. The week profile is compared with the fourth week measurement in Figure 4.18. We can see in the figure that the error between the week profile and the fourth measurement is still relatively large. As we explained above, the relative error will be used to predict the traffic flow. Figure 4.19 shows the relative error. The relative period errors are relatively small, most relative errors are between -1.5 and 1.5. The reader should note the memory present in the relative period errors which are taken care of in the short term prediction. Performance of the short term prediction based on the relative error will be explained in the following subsection.

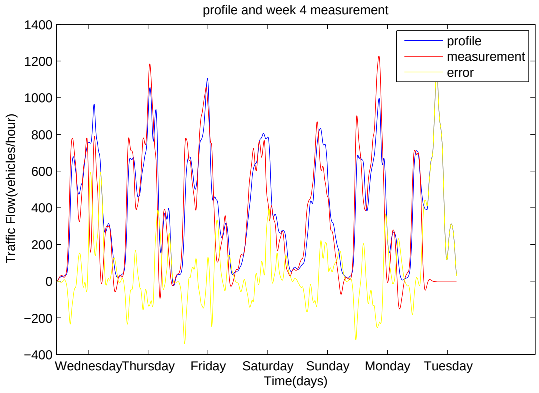


Figure 4.18: Errors between week profile and last week measurement.

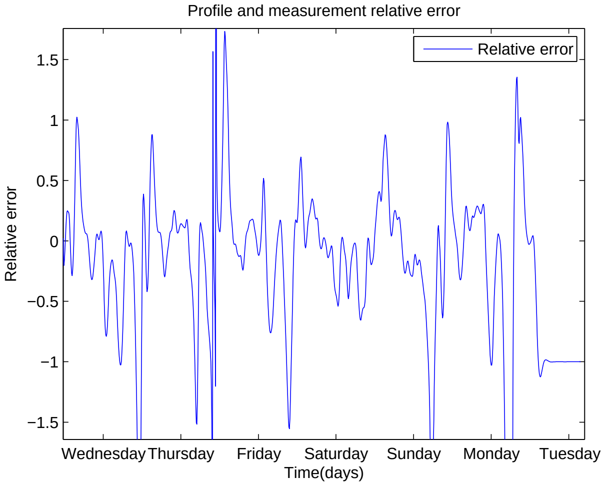


Figure 4.19: Relative errors between week profile and last week measurement

The short term predictions of the relative errors

The relative errors are predicted based on the equations of subsection 4.4.3.

The short term predictions of the traffic flows

Finally the prediction of the traffic flow is calculated based on the equations of subsection 4.4.3. The predictions of different days can be seen in Figure 4.21, Figure 4.20, Figure 4.22, and Figure 4.23. As we can see from these figures, the traffic flow can be well predicted for on-ramp traffic.

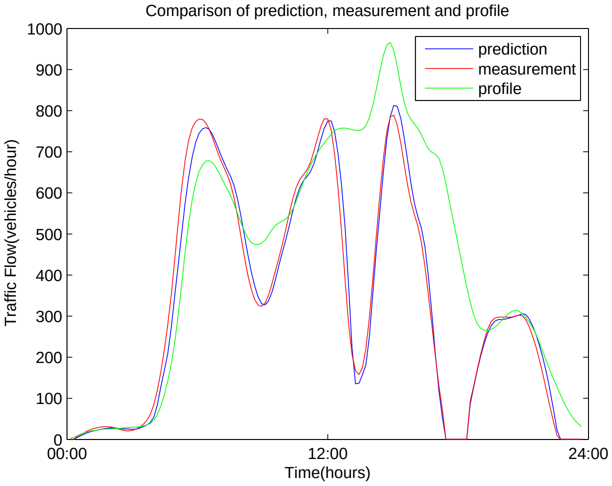


Figure 4.20: Adaptive predictions of Wednesday's traffic flow.

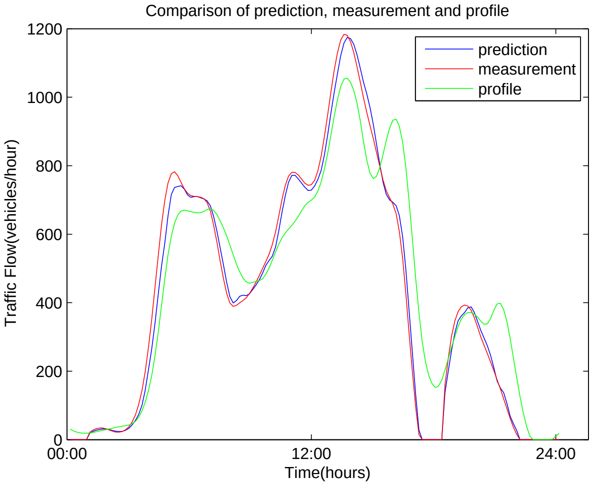


Figure 4.21: Adaptive predictions of Thursday's traffic flow.

Figure 4.24 shows 3-step predictions on Monday from 10:00 to 12:00. The circles represent the week profile, the squares represent the current measurement. The . , x and * represent the first step prediction, the second step prediction and the third step prediction, respectively. The red lines display on the three steps predictions. We can see from the figure that the three steps predictions are much closer with the measurements compared with the profile.

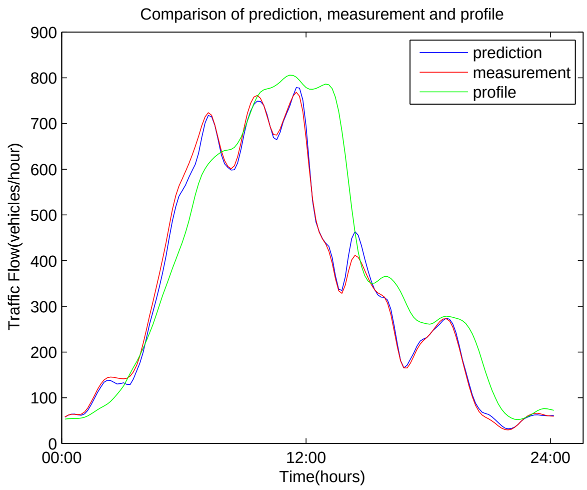


Figure 4.22: Adaptive predictions of Saturday's traffic flow.

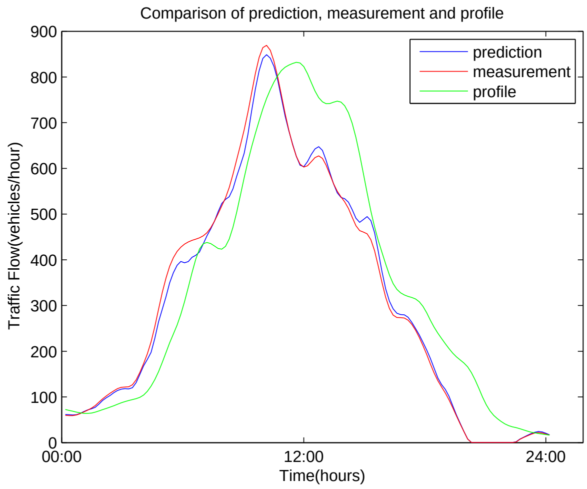


Figure 4.23: Adaptive predictions of Sunday's traffic flow.

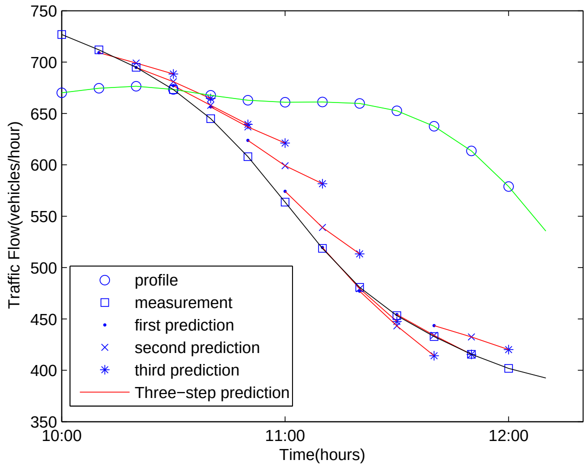


Figure 4.24: Three-Step traffic flow predictions. one step corresponds to 10 minutes.

Table 4.5: Comparison of the adaptive predictor and of the week profile.

	1st step prediciton(SP)	2nd SP	3rd SP	Week profile
RMSE	40.9	62.2	167	227.8
RMPE	10.8	20.7	33.0	3.3e+5
VAF	98.4	96.3	73.0	54.6

Table 4.6: Comparison after the improvements.

	1st step prediciton(SP)	2nd SP	3rd SP	Week profile
RMSE	32.7	40.6	65.6	225.5
RMPE	7.5	11.1	19.8	2.8e+4
VAF	98.9	98.4	95.8	54.5

4.5.11 Performance of the algorithm in case of on-ramps traffic flow

The results of the demand prediction by using the adaptive filter are compared with the week profile. Table 4.5 shows the RMSE, RMPE and VAF values of the adaptive filter and the week profile. We can conclude that the adaptive filter can better predict the traffic flow than the week profile from on-ramps data. Table 4.6 shows the RMSE, RMPE and VAF values of the adaptive filter and the week profile after the constraint described in section 4.5.6 is applied. We can see that there is almost no change for the first and the second step prediction, however there is a big improvement for the third step prediction.

4.6 Conclusions

This chapter provides an adaptive algorithm for prediction of traffic flow at entry points to a motorway network. The algorithm is based on an existing algorithm which was adjusted for the particular problem at hand and illustrated with the data of the ring road of Amsterdam. The results presented in the chapter show that the algorithm provides robust predictions with relatively small error and fast computation (in the order of 0.01 second) for the next 30 minutes of traffic flow, both at motorway entries and at on-ramps.

Chapter 5

On-line Distributed Traffic Prediction[1]

5.1 Introduction

The usual approach to traffic prediction is a network-based model that simulates the traffic based on spatial information, temporal information and traffic flow dynamics. There are many different types of network-based models in the literature such as microscopic, mesoscopic, and macroscopic traffic models. The network-based prediction model can be used as a decision support system to evaluate the impact of different control scenarios for the purpose of both off-line scenario planning and on-line scenario selection.

However, centralized prediction of traffic flow is not suitable for simulation and for scenario evaluation of large networks such as all motorways in the Netherlands since long computation times and other drawbacks make it unsuitable for real-time use. Thus, an algorithm is needed for distributed/parallel prediction of traffic flow.

The purpose of this chapter is to present an algorithm for the distributed/parallel prediction of a large-scale motorway network including rings and to describe the performance of the algorithm for a motorway network near Amsterdam.

The contents of the chapter is as follows. The next section contains the problem statement. The framework of the proposed approach is described in section 5.3. The performance evaluation criteria of the proposed algorithm is discussed in section 5.4. A case study is presented to illustrate the performance of the proposed approach in section 5.5. Concluding remarks can be found in section 5.6. The appendices provide information about technical details of the algorithm.

[1] This chapter is based on my publications Wang et al. (2014b, 2010a, 2012a)

5.2 Problem Statement

The motivation of this research has been stated in chapter 4. To assist the operators with their control tasks, predictions of traffic flow have to be generated for a prediction horizon. The predictions are to be used for control of motorway traffic, in particular to evaluate the current state of the control measures and to evaluate the performance of one or more control scenarios.

The problem treated in this chapter is to predict the traffic flow in a large-scale motorway network for the next 30 minutes within a time frame of seconds. To solve the problem, one has to exploit the computational power of multiple cores or multiple PCs in order to meet the performance objective of a short computation time.

5.3 The Framework for Distributed Prediction

The proposed approach of the distributed or parallel prediction/simulation consists of the following steps:

1. Partition the full network into two or more subnetworks each of which has only an one-directional traffic flow;
2. Predict the traffic flow of each subnetwork based on an initialization of the predicted inflow of the subnetwork;
3. Take care of the consistency of the predictions produced for the subnetworks; thus if the outflow of one subnetwork is to be equal to the inflow of another network then take care that these traffic flows are approximately equal. The consistency requires an algorithm with iterations.

The detailed description of each of the above steps follows in the next subsections.

In Section 5.5 the example of the ring network of Amsterdam is used. A realistic system will be formulated and distributed prediction will be proposed. To illustrate the theoretical framework of the chapter a simple ring road network will be used.

Example 5.3.1 *Consider a road network consisting of a ring. The ring road has traffic inflows from motorways and traffic inflows from on-ramps. The example is continued later on in this section.*

5.3.1 Partitioning of the network

Any road network will be partitioned into two or more subnetworks. The main objective for such a partitioning of a road network is to reduce computation time by the synchronous parallel computations of the subnetworks. The selection of the number of subnetworks and the size of subnetworks is a complexity problem which requires further research. It is briefly discussed in the subsection.

Consider then the problem of partitioning or decomposing a road network into two or more subnetworks. The criterion for such a partition is to allow distributed and parallel computation and to achieve an approximately equal load for the computations of the subnetworks. In general, decomposition approaches include functional decomposition, temporal decomposition and spatial decomposition. Spatial decomposition is the most common approach in the field of road traffic simulation. The existing spatial decomposition approaches are equally-sized method Klefstad et al. (2005), minimization of Neighbors Nagel & Rickert (2001), Macroscopic-Simulation-Based Division (MaSBD) and Microscopic-Simulation-Based Division(MiSBD) Potuzak (2011).

Below a road network is related to a graph. The nodes of the graph correspond to particular locations of the road network, often an intersection or a crossing. An edge of a graph corresponds in the road network to a road from one location to an adjacent location. If one distinguishes the driving directions of each road then every edge is associated with that direction and the graph is then called a directed graph. A path in a directed graph is a sequence of directed edges. A tree in the graph is defined as a node from where exists a directed path to all other nodes. This concept is used below.

Consider then a network of motorways possibly with several rings. The problem is to partition the considered motorway network into two or more *subnetworks* such that the following requirements are met.

Definition 5.3.2 *Define the* partitioning requirements *as:*

1. *No subnetwork has a ring of roads;*
2. *The traffic flow in any subnetwork is unidirectional;*
3. *A boundary point of a subnetwork should not be within any network bottleneck.*
4. *The length of a subnetwork is in the range of 9 to 16 km; the upper bound is a somewhat arbitrary choice motivated by the duration of the computation time of the full prediction recursion.*
5. Load balancing. *The computation times of the subnetworks should be approximately equal.*
6. *The number of interconnections between subnetworks should be minimal.*

Problem 5.3.3 Partition *a motorway network into several subnetworks such that each subnetwork meets the partitioning requirements of Def. 5.3.2.*

The preliminary procedure to partition a large motorway network into two or more subnetworks is stated below.

Procedure 5.3.4 *Consider a motorway network.*

1. *If the network or if a subnetwork contains a ring then cut the ring into two or more subnetworks such that each subnetwork consists of a stretch of motorway and no stretch contains a motorway intersection in the interior of the subnetwork. Thus the ring is first cut at the points where there are motorway intersections.*

2. *If the remaining collection of subnetworks contains a subnetwork which is longer than the maximum stretch length set in the partitioning criteria then cut the subnetwork into pieces of almost the same length such that each new subnetwork meets the partitioning criteria.*

The argument to cut the ring network at the junction with the motorways is a choice which is motivated by experience with larger networks. It allows a simple partitioning in case of networks with two or more rings. Experience has to be gained with this choice of partitioning. The approach is illustrated in the case study in Section 5.5. For the road network of the province of North-Holland in The Netherlands, there will be approximately 100 subnetworks.

Example 5.3.5 *Consider the simple ring network of Example 5.3.1. The ring is partitioned in two parts. See Figure 5.1 for the illustration of the partition.*

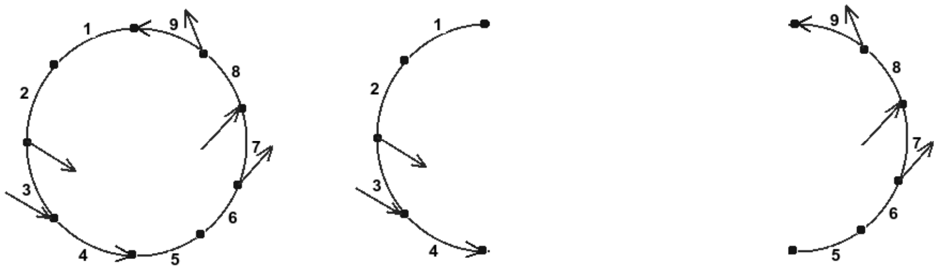


Figure 5.1: The simple ring network.

5.3.2 Prediction of the Traffic Flow in a Subnetwork

Consider then a subnetwork as obtained by the partitioning approach of Subsection 5.3.1. The subnetwork is assumed to satisfy the partitioning conditions of Def. 5.3.2.

Problem 5.3.6 Prediction of the traffic flow in a subnetwork. *Predict the traffic flow for the considered prediction horizon in each of the cells of the subnetwork and of the outflows of the network. It is assumed that there are available predictions of all the inflows of the subnetwork, i.e. those from other motorways and from on-ramps.*

It is assumed that there is available a system describing the traffic flow in the subnetwork. The prediction of the traffic flow in a deterministic system is based on simulation of the system of the traffic flow in the subnetwork. There are various types of models for the dynamics of traffic flow such as microscopic, mesoscopic and macroscopic traffic models Smulders (1986); Messmer & Papageorgiou (1990); Lighthill & Whitham (1955); van Lint et al. (2008); Wang et al. (2010a). In this chapter, a macroscopic traffic model is used to demonstrate the proposed parallel simulation approach. A macroscopic traffic model computes per cell, based on predictions of all traffic inflows into the subnetwork, the predictions of the density and of the speed of the traffic flow in the subnetwork and of all traffic outflows to other subnetworks.

Each subnetwork must have boundary points. At these boundary points there are traffic inflows and outflows. The traffic inflows include: (1) inflows of one or more motorways into the subnetwork and (2) the inflows from all on-ramps. If the origin of a subnetwork is a traffic merge then there will be traffic inflows from two motorways. If the origin of the subnetwork is at a motorway intersection there could be three traffic inflows from as many motorways entering a subnetwork. The traffic outflows include: (1) the motorway traffic outflows of the subnetwork to other subnetworks; (2) the traffic outflows of off-ramps. Note that at a traffic intersection there could be two or three motorway outflows. For the remainder of the chapter it is necessary to distinguish between the motorway inflows and the other traffic inflows. The prediction model must have the information of all inflows at these boundary points for the prediction horizon.

An adaptive prediction algorithm for the traffic inflows of a subnetwork at boundary points of the network, both for motorways and for on-ramps has been proposed in chapter 4. The purpose of the adaptive prediction algorithm is to provide predictions of traffic flow at the boundary points of the road network. The effectiveness of these adaptive predictions was illustrated in those references for data of traffic flow on the Amsterdam ring network.

Below a system is defined which describes the traffic flow in a subnetwork. Such models are standard in the literature of dynamic traffic management and such models are described in a previous chapter. A macroscopic model is used to describe the traffic flow. The road network of a subnetwork is partitioned into several sections, typically of 500 m. length. Each section has two state variables, the density of the traffic in vehicles per km per lane and the speed of the traffic in the section in km per hour. The dynamics of the system is not detailed in this chapter, it is denoted by the function f in Def. 5.3.7. The reader may find the METANET model in Messmer & Papageorgiou (1990) and other models in Smulders (1987a); J.H. van Schuppen & de Waal (1999). In Section 5.5 for the actual simulations shown, use was made of a linearized system and of the Fast Lane model, see van Lint et al. (2008).

Definition 5.3.7 *Define the* system for the traffic flow in a subnetwork *denoted by the equations,*

$$\begin{aligned} x(t+1) &= f(x(t), u_m(t), u_{in}(t)),\ x(t_0) = x_0, \\ z_m(t) &= h_m(x(t)), \\ & T = \{t_0, t_0+1, \ldots, \} \subset \mathbb{Z},\ \textit{the time index set}, \\ & x : T \to \mathbb{R}^n,\ u_m : T \to \mathbb{R}^{m_m},\ u_{in} : T \to \mathbb{R}^{m_{in}},\ z_m : T \to \mathbb{R}^{p_z}. \end{aligned}$$

In these equations, the vector $x(t)$ represents the state with a density and a speed for each section of the subnetwork, the input u_m represents the traffic inflow(s) of motorways, and the input u_{in} represents the traffic inflows of on-ramps. The variable z_m represents the traffic outflow of the subnetwork to motorways, which is the product of the density and the speed of the last section of the subnetwork. The off-ramp traffic flow is not explicitly indicated, since we assume that it is a certain percentage of traffic flow on motorways.

For control of motorway networks the above system can be generalized to a control system with inputs for the traffic control measures.

The prediction system produces on-line predictions of the traffic flow in the subnetwork. This is done sequentially, for any time the system produces predictions for the prediction horizon. For example, at 10:00 hours it produces predictions for the horizon 10:00 to 10:30 hours, and at 10:10 for the horizon 10:10 to 10:40 hours, etc.

Definition 5.3.8 *Define* the horizon-prediction system of a subnetwork *in terms of its inflows by the following equations. Fix a time moment $t \in T$. Of the motorway input denoted by u_m denote the prediction of the value at time $t+1$ made at time $t \in T$ by $\hat{u}_m(t+1|t)$, for subsequent times by $\hat{u}_m(t+s|t)$ for $s \in \mathbb{Z}_+$, and similarly for the other variables.*

The predictions of the traffic outflow are computed by simulation of the system, see the equations,

$$\begin{aligned} \hat{x}(t+s+1|t) &= f(\hat{x}(t+s|t), \hat{u}_m(t+s|t), \hat{u}_{in}(t+s|t)),\ \hat{x}(t+1|t) = \hat{x}(t), && (5.1) \\ \hat{z}_m(t+s|t) &= h_m(\hat{x}(t+s)),\ s = 1, 2, \ldots, t_p, && (5.2) \\ & \hat{x}(t+s|t) \in \mathbb{R}^n,\ \hat{u}_m(t+s|t) \in \mathbb{R}^{m_m},\ \hat{u}_{in}(t+s|t) \in \mathbb{R}^{m_{in}}. && (5.3) \end{aligned}$$

The initial state of the predictor, $\hat{x}(t)$, is the one-step prediction produced by a state estimator. This estimate is produced by what is called the filter system. The filter system estimates the state of the traffic system for all sections and for all state variables. A state estimator was formulated in the publications of S.A. Smulders, see Smulders (1987b, 1988, 1990). It is assumed that a state estimator is available in the traffic control center of the subnetwork.

The variable $\hat{x}(t+s|t)$ denotes the prediction of the state $x(t+s)$ based on observations available at time t. Similarly, the variable $\hat{z}_m(t+s|t)$ denotes the prediction of the *traffic outflow of the subnetwork to motorways at time $t+s$ based on observations available at time t. These outflows may be inflows of another subnetwork or they may leave the considered road network.*

Denote the predictions of several variables on a prediction horizon by,

$$\begin{aligned}
& && t \in T, \text{ a time moment of the time index set,} \\
T_p(t) &=&& \{t+1, t+2, \ldots, t+t_p\}, \\
& && \text{the prediction horizon at time } t \text{ of length } t_p \in \mathbb{Z}_+, \\
\hat{u}_m(t+1:t+t_p|t) &=&& \left(\ \hat{u}_m(t+1|t),\ \ \hat{u}_m(t+2|t),\ \ \ldots\ \ \hat{u}_m(t+t_p|t)\ \right)^T \in \mathbb{R}^{t_p m_m}, \\
& && \text{the predicted inflows of motorways on the prediction horizon,} \\
\hat{u}_{in}(t+1:t+t_p|t) &=&& \left(\ \hat{u}_{in}(t+1|t),\ \ \hat{u}_{in}(t+2|t),\ \ \ldots\ \ \hat{u}_{in}(t+t_p|t)\ \right)^T \in \mathbb{R}^{t_p m_{in}}, \\
& && \text{the predicted inflows of on-ramps,} \\
\hat{z}_m(t+1:t+t_p|t) &=&& \left(\ \hat{z}_m(t+1|t),\ \ \hat{z}_m(t+2|t),\ \ \ldots\ \ \hat{z}_m(t+t_p|t)\ \right)^T \in \mathbb{R}^{t_p p_z}, \\
& && \text{the predicted outflows to motorways on the prediction horizon.}
\end{aligned}$$

The map defined above by the algorithm which produces from an estimate $\hat{x}(t)$ of the state of the system at time $t \in T$, from the predicted traffic inflows of motorways $\hat{u}_m(t+1:t+t_p|t)$ and of on-ramps $\hat{u}_{in}(t+1:t+t_p|t)$ on the prediction horizon, the predicted outflows of the subnetwork over the prediction horizon $\hat{z}_m(t+1:t+t_p|t)$ is denoted by,

$$\begin{aligned}
\hat{z}_m(t+1:t+t_p|t) &= f_h(\hat{x}(t), \hat{u}_m(t+1:t+t_p|t), \hat{u}_{in}(t+1:t+t_p|t)), \qquad (5.4) \\
& \quad f_h : \mathbb{R}^n \times \mathbb{R}^{t_p m_m} \times \mathbb{R}^{t_p m_{in}} \to \mathbb{R}^{t_p p_z}.
\end{aligned}$$

Several of the components of the predicted outflows of a subnetwork are equal to the predicted inflows of a downstream subnetwork as will be described below.

Typically, the time step of the discrete-time system is 10 s. while the prediction horizon is 30 minutes, and the dimension of the motorway input is 1 (one), hence there are $30 \times 6 = 180$ time steps of the prediction horizon. With these numbers, the dimensions of the vectors $\hat{u}_m(t+1:t+t_p|t)$ and $\hat{z}_m(t+1:t+t_p|t)$ are then 180.

5.3.3 Subnetworks predictions made consistent at the Network Level

How to obtain predictions computed by the subnetworks which are consistent at the network level?

The framework for prediction of traffic flow in a network consists of a two level system. At the subnetwork level there are two or more subnetworks. The subnetworks have been obtained by partitioning the network into several parts. As described in the previous subsection, each subnetwork is able to compute from predicted inflows of the subnetwork over the prediction horizon the predicted outflows of the subnetwork over the same prediction horizon. Those computations can be carried out in parallel reducing complexity. But the output of a subnetwork may have to be equal to the input of another subnetwork and this may hold for a large number of subnetworks.

At the network level therefore the predictions of the subnetworks have to be made consistent. The task of the prediction system at the network level is to produce predictions of the subnetworks of which the traffic outflows of a subnetwork are equal or approximately equal to the traffic inflows of a downstream subnetwork if the subnetworks are so related. The approximation criterion for the corresponding traffic outflows and traffic inflows will be specified below.

Problem 5.3.9 Problem of consistent predictions at the network level. *Consider a motorway network and a prediction system with two levels, the lowest level for the predictions of the subnetworks and the highest level for the prediction of the network. Formulate a procedure for obtaining predictions of the traffic flow in the network on the prediction horizon such that the motorway traffic outflows of a subnetwork are equal to or approximately equal to the motorway traffic inflows of the corresponding downstream subnetwork if there exists such a downstream network.*

The procedure to achieve consistency is to be based on an approximation criterion. The approximation criterion compares (1) the traffic inflows into a subnetwork with (2) the predictions of traffic outflows of the corresponding upstream subnetwork for the same prediction horizon and for the same locations and that for all locations of the network where two subnetworks are connected. If inflows and outflows are not consistent, a minimal demand supply scheme van Lint et al. (2008) has to be used to determine inflows and outflows.

Definition 5.3.10 *Define the* horizon-prediction system of a network *by the following objects and functions.*

Denote the index set of all subnetworks of the considered network by the set $J = \{1, 2, \ldots, j_{sn}\}$. *For all* $j \in J$ *define the index subsets of subnetworks,*

$$J_{in}(j) = \{i \in J | \exists \textit{ motorway outflow of Subnetwork } i \textit{ to Subnetwork } j,\} \quad (5.5)$$

$$J_{out}(j) = \{i \in J | \exists \textit{ motorway outflow of Subnetwork } j \textit{ to Subnetwork } i.\} \quad (5.6)$$

The indices of the system of Subnetwork $j \in J$ *are now denoted by,* $n_{net,j}$ *the state-space dimension,* $m_{net,j,m}$ *the dimension of the motorway input,* $m_{net,j,in}$ *the dimension of the external inflows, and* $p_{net,j,z}$ *the dimension of the motorway outflow.*

Recall that for any subnetwork $j \in J$ *the horizon-prediction system of Subnetwork* j *at time* $t \in T$ *based on a prediction of the traffic inflow over the prediction horizon from outside the network denoted by the input vector* $\hat{u}_{j,m}(t+1 : t+t_p|t)$ *produces the traffic* *outflows over the prediction horizon as the vector* $\hat{z}_{j,m}(t+1:t+t_p|t)$*. Denote then,*

$$\begin{aligned}
m_{mp} &= t_p \sum_{j=1}^{j_{sn}} m_{net,j},\; m_{inp} = t_p \sum_{j=1}^{j_{sn}} m_{net,j,in},\; p_{zp} = t_p \sum_{j=1}^{j_{sn}} p_{net,j,z}, && (5.7)\\
\hat{u}_{net,m}(t+1:t+t_p) &= \begin{pmatrix} \hat{u}_{net,1,m}(t+1:t+t_p|t) \\ \vdots \\ \hat{u}_{net,j_s,m}(t+1:t+t_p|t) \end{pmatrix} \in \mathbb{R}^{m_{mp}}, && (5.8)\\
& \textit{correspondingly,}\\
& \hat{u}_{net,in}(t+1:t+t_p|t) \in \mathbb{R}^{m_{inp}},\; \hat{z}_{net,m}(t+1:t+t_p|t) \in \mathbb{R}^{p_{zp}}.
\end{aligned}$$

The graph of the network and its partition into subnetworks defines the relation between outflows of a subnetwork to inflows of the corresponding downstream subnetwork. Thus, the outflow of traffic of a subnetwork j may equal the inflow of traffic of another subnetwork, say i. However, there are subnetworks whose inflow is only the traffic flow from outside the network. For the k-th iteration of the computation the relation between the traffic flows can then be described by the equation,

$$\hat{u}_{net,i,m}^{(k+1)}(t+1:t+t_p|t) = \begin{cases} \hat{z}_{net,j}^{(k)}(t_1:t+t_p|t), & \textit{if } i \in J_{out}(j), \\ \hat{u}_{net,i,m}(t+1:t+t_p|t), & \textit{if } i \in J \backslash J_{out}(j). \end{cases} \tag{5.9}$$

This relation is formalized in the linear map,

$$\begin{aligned}
& F_{net,z} \in \{0,1\}^{m_{mp} \times p_{zp}},\; F_{net,m} \in \{0,1\}^{m_{mp} \times m_{mp}},\\
F_{net,z,i,j} &= \begin{cases} I_{t_p}, & \textit{if } i \in J_{out}(j), \textit{ or, } \hat{u}_i(t+1:t+t_p|t) = \hat{z}_j(t_1:t+t_p|t), \\ 0 & \textit{else;} \end{cases} && (5.10)\\
F_{net,in,i,j} &= \begin{cases} I_{t_p}, & \textit{if } i \in J \backslash J_{out}(j), \textit{ or}, \hat{u}_i(t+1:t+t_p|t) = \hat{u}_{i,in}(t+1:t+t_p|t), \\ 0 & \textit{else;} \end{cases} && (5.11)
\end{aligned}$$

$$\begin{aligned}
\hat{u}_{net,m}^{(1)}(t+1:t+t_p|t) &= F_{net,z}\hat{z}_{net}(t+1:t+t_p|t) + F_{net,in}\hat{u}_{net,in}(t+1:t+t_p|t),\\
&= F_{net,z}f_{net}(\hat{x}(t), \hat{u}_{net,m}^{(0)}(t+1:t+t_p|t), \hat{u}_{net,in}t+1:t+t_p|t)) +\\
&\quad + F_{net,in}\hat{u}_{net,in}(t+1:t+t_p|t), && (5.12)\\
&\quad \textit{where the index } (0) \textit{ denotes the initial value and}\\
&\quad (1) \textit{ denotes the first iteration.}
\end{aligned}$$

The matrices $F_{net,z}$ and $F_{net,in}$ consists of zeros and ones only and are based only on the graph of the network. In case the outflow of a subnetwork is not an input to another subnetwork, then the vector $\hat{u}_{net}^{(1)}(t+1:t+t_p|t)$ equals for that component the predicted traffic inflow produced by the adaptive filter and denoted by $\hat{u}_{net,m}(t+1:t+t_p|t)$.

Example 5.3.11 *Consider the simple ring network, see Example 5.3.1 and Example 5.3.5. For this ring network with only two subnetworks and without external inflows from motorways the matrices of the interconnection map are*

$$F_{net,z} = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix},\; F_{net,in} = 0.$$

Next the consistency of the input predictions and of the computed outputs can be defined mathematically.

Definition 5.3.12 *Define the* fixpoint equation of consistency of traffic flows *at the network level as stated below. Consider the horizon-prediction system at the network level of Def. 5.3.10. Consider a prediction time $t \in T$. In reference to this system define the* fixpoint equation of consistency *as*

$$\begin{aligned}
\hat{u}^{(0)}_{net,m}(t+1:t+t_p|t) &= \hat{u}^{(1)}_{net,m}(t+1:t+t_p|t), \\
&= F_{net,z}(f_{net}(\hat{x}(t), \hat{u}^{(0)}_{net,m}(t+1:t+t_p|t), \hat{u}_{net,in}t+1:t+t_p|t)) + \\
&\quad +F_{net,in}\hat{u}_{net,in}(t+1:t+t_p|t) && (5.13) \\
&= f_{fp}(t, \hat{u}^{(0)}_{net,m}(t+1:t+t_p|t)), \textit{ hence,} && (5.14) \\
v &= f_{fp}(t,v), \textit{ is the fixpoint equation and then} && (5.15) \\
v &= \hat{u}^{(0)}_{net,m}(t+1:t+t_p|t) \in \mathbb{R}^{m_{mp}}. && (5.16)
\end{aligned}$$

In the fixpoint equation, the state at time $t \in T$, $\hat{x}(t)$, and the inflows of on-ramps, $\hat{u}_{net,in}(t+1:t+t_p|t)$, are fixed to the values provided. The symbol f_{fp} denotes the function of the fixpoint equation.

In B.2, two methods are proposed to solve the fixpoint equation: a successive approximation method and a recursive method. After a comparison of the two methods, the distributed successive approximation algorithm is chosen for this chapter. We prove that the distributed successive approximation algorithm for the approximation of the solution of the fixpoint equation is quite effective and converges sufficiently fast for the application considered.

In B.3, the fixpoint method is analyzed for the simple ring network with a linear system. We prove that the conditions for the existence of a unique fixpoint of Equation (B.12) are satisfied. The convergence speed is determined by the eigenvalues of the system.

In B.1, the complete prediction algorithm is stated.

In realistic models of traffic flow the dynamics of the system is nonlinear. Then the questions of existence of a solution to the fixpoint equation, the uniqueness of the solution, and the convergence rate are not so easy to answer.

Based on the structure of the system, one can argue that the system should be contractive, meaning that there exists a solution of the fixpoint equation. The reason for this is that the dynamic system is dissipative as understood in control theory. The traffic flow passing at any point in the subnetwork leaves the network at all off-ramps hence if one considers a ring then after one loop no cars who have entered at this point will exit the network at this point.

The domain of attraction of the unique solution is expected to be the full state set. The convergence rate is not clear because of the nonlinearities of the system. The high dimensions of the system will make it difficult to carry out a mathematical analysis of the convergence.

In case of congested traffic the convergence rate could be much slower than in noncongested traffic.

5.4 Performance

In this chapter, the performance of the proposed algorithm will be evaluated in terms of computational complexity and communication complexity.

5.4.1 Communication Complexity

For the communication protocol for sending a prediction of traffic outflow, say $\hat{z}_{net,j}(t+1:t+t_p|t)$, from subnetwork $j \in J$ to a downstream subnetwork $i \in J$ there are two options:

1. *Communication protocol 1.* There exists a dedicated communication line from the controller of Subnetwork $j \in J$ to the controllers of its downstream Subnetworks $i \in J_{out}(j)$ and that for all $j \in J$, which can be used without any interference with communications between other subnetworks;
2. *Communication protocol 2.* There exists only communication lines between any subnetwork and the network communication center of the full network. In this case a subnetwork whose prediction vector of traffic outflows is needed by another subnetwork communicates its prediction vector to the network communication center which in turn forwards this prediction vector to the downstream subnetworks needing that prediction vector.

The author of this thesis favors Communication Protocol 1 because of its much lower communication complexity. The communication complexity differs by the communication protocol used. The complexity formula is derived in Subsection 5.4.2. It is clear that the first communication protocol has a much lower communication time complexity than the second communication protocol. Therefore, the road operator has an incentive to install dedicated communication lines between adjacent subnetworks. This involves economic costs which are not treated in this chapter.

Next the second aspect of distributed computation of the vector of inflow traffic predictions is discussed. There are two ways to handle the computations for the approximation criterion:

1. The approximation criterion is computed at the network level.

2. The approximation criterion is computed at each of the subnetworks and only whether the criterion is met or not is communicated to the network level.

The approximation criterion of the recursive computation of the solution of the fixpoint equation is often a vector norm. At iteration $(k+1) \in \mathbb{Z}_+$ the approximation criterion decomposes over the subnetworks as displayed in the next equation for the L_2 vector norm,

$$\begin{aligned} \|\hat{u}^{(k+1)}_{net,m}(t+1:t+t_p|t) - \hat{u}^{(k)}_{net,m}(t+1:t+t_p|t)\|_2^2 &= \\ \sum_{j=1}^{n_{net}} (\varepsilon^{(k+1)}_{net,j} c_j)^2 / \sum_{j=1}^{n_{net}} c_j,\ c_j = \dim \varepsilon_{net,j}. \end{aligned}$$

For the l_1 norm of a vector in $\mathbb{R}^n$ the sum operation is replaced by a maximization operation. It is of interest for the efficient computation that an approximation criterion is selected which allows a distributed computation of the criterion.

The conclusion is thus that the computation of the approximation criterion for the network can be distributed to computation of the approximation criterion for each subnetwork combined with communication of the local norms to the network communication center. This is a gain in communication complexity because each subnetwork only sends one real number, the value of its local approximation criterion. The communication protocol is then that each subnetwork (1) computes the value of the local approximation criterion, say

$$\varepsilon^{(k+1)}_{net,i} = \|\hat{u}^{(k+1)}_{net,i,m}(t+1:t+t_p|t) - \hat{u}^{(k)}_{net,i,m}(t+1:t+t_p|t)\|_2,$$

and communicates this real number to the communication center of the network. At the network level the value of the approximation criterion of the network is computed. (2) the network communication center communicates the outcome of the network approximation criterion to all subnetworks.

The second method for computation of the approximation criterion is that the computations are done entirely at the subnetwork level. Thus, the subnetworks could compute the value of the subnetwork approximation criterion and compare this to the threshold $\varepsilon_{net,i,m} = \varepsilon_{net,m}/n_{net}$ where $\varepsilon_{net,m}$ denotes the tolerance level at the network. The threshold then is uniformly distributed over all subnetworks. The fact that at a subnetwork the approximation criterion is met is then communicated from each subnetwork to the network. Once the network prediction system has received a reaction from all subnetworks that the prediction approximation criteria are met, it informs all subnetworks that the approximation for the full network has been achieved.

Which of these methods is better than the other, is not yet clear. The decision is based on the computational and the communication complexity.

5.4.2 Computational Complexity

The time complexity of the computation is of interest because of the distributed character.

Consider the distributed computation of an approximation of the solution of the fixpoint equation. This complexity is to be compared with a centralized computation. Below the formula for the time complexity follows. Because only time complexity is considered and space complexity is not discussed, the expression *complexity* denotes from now on the time complexity.

Proposition 5.4.1 *Consider the algorithm for the distributed computation of an approximation solution of the fixpoint equation.*

(a) The complexity of the distributed computation (acronym comp) and that of the communication (acronym comm) with Communication Protocol 1 is polynomial and equals,

$$O_{comp}((n^* + p_z^*)t_p k_{max}), \ \ O_{comm}(m_{mp}^* t_p k_{max}). \tag{5.17}$$

(b) The complexity of the distributed computation and of the communication with Communication Protocol 2 is polynomial and equals,

$$O_{comp}((n^* + p_z^*)t_p k_{max}), \ \ O_{comm}(j_{sn} m_{mp}^* t_p k_{max}). \tag{5.18}$$

(c) The complexity of the computation by a centralized algorithm is polynomial and equals,

$$O_{comp}((n^* + p_z^*)t_p j_{sn} + m_{mp}^* t_p j_{sn}), \ \ O_{comm} = 0. \tag{5.19}$$

Distributed computation has thus a lower complexity than centralized computation if $k_{max} < j_{sn}$, thus if the maximal number of iterations required for the approximation is smaller than the number of subnetworks.

The symbols denote,

t_p — *the length in steps of the prediction horizon,*

k_{max} — *the maximal number of iterations of the fixpoint approximation,*

n^* — *the maximum of the state-space dimensions of all subnetworks,*

$$n^* = \max_{j \in J} n_{net,j},$$

m_{mp}^* — *the maximum of the dimensions of the motorway inflows,*

$$m_{mp}^* = \max_{j \in J} m_{net,j,m},$$

p_z^* — *the maximum of the output dimensions of all subnetworks,*

$$p_z^* = \max_{j \in J} p_{net,j,z},$$

j_{sn} — *the number of subnetworks of the road network.*

Proof (a) Consider first the computation of the predictions by a subnetwork. For each step of the prediction a state of dimension n and an output of dimension p_z have to be computed, hence the term $(n+p_z)$. This has to be done for t_p time steps of the prediction horizon, hence $(n+p_z)t_p$. Because the computations are done in parallel for all subnetworks, the expression $(n^*+p_z^*)t_p$ follows. According to Communication Protocol 1, there is the communication time for the communication of the vector $\hat{u}_m(t_1 : t+t_p|t)$ from a subnetwork to a downstream network. The time complexity of this communication is thus $m_{mp}^* t_p$ because the communications are carried out in parallel.
(b) The difference with part (a) is that the communications are now carried out in a centralized way hence in series for all subnetworks. Hence the communication complexity is $j_{sn} m_m^* t_p$, and that two times which factor 2 is not listed.
(c) The formula follows analogously as in (a) and (b) but with differences. The system has now a state space dimension which is the sum of all state-space dimensions of the subnetworks, which is upperbounded by $j_{sn} n^*$. The second term in the complexity formula is because the end result has to be communicated to the subnetwork in by a centralized protocol.

In an example, typical values for the parameters of the complexity formulas are $t_p = 180$, $n = 1$, $m_{mp}^* = 1$, $p_z = 1$, $j_{sn} = 10$. Distributed computation has then a lower time complexity if the maximal number of iterations is lower than $j_{sn} = 10$.

5.5 Case Study: distributed prediction in AgentScape

The performance of the proposed prediction procedure is best evaluated based on actual on-line traffic data. At the time this chapter is written, the author of this thesis do no longer have access to on-line traffic data. The hope that in the near future a company with access to on-line traffic data is willing to test the proposed prediction algorithm with actual data.

The reader finds below an evaluation of the prediction procedure for a medium-scale road network, about 144 km. in length. Simulations of the prediction of the traffic flow over a horizon of 30 minutes are provided. In addition, the computational time complexity is summarized.

5.5.1 Introduction: AgentScape

AgentScape is a middleware layer that supports large-scale agent systems. It has been implemented in Java by the Faculty of Technology, Policy and Management of the Delft University of Technology (TUD). The rationale behind the design decisions are

1. to provide a platform for large-scale agent systems,

2. support multiple code bases and operating systems,

3. interoperability with other agent platforms.

The main motivation of using AgentScape in the case study is that it provides a platform to simulate agents in multi-cores or multi-PCs. It has the functionality of assigning the resources of CPUs according to the demanding simulation/prediction tasks of different agents automatically. It also has the functionality to handle the communications between two agents. Such that researchers or developers only need to focus on implementing the simulation/prediction algorithm without thinking about the software architecture or the low-level communication software.

In the case study, both a linear model and a nonlinear model will be implemented in AgentScape to demonstrate the proposed algorithm. Each agent represents a subnetwork. For the linear model experiment, two agents are used to represent two subnetworks of the toy ring road; For the nonlinear model experiment, four agents are used to represent the four subnetworks of the A10 inner ring to demonstrate the parallel prediction algorithm. The case study is performed under Dell 4 Core i7 Windows 7. For maximal usage of the CPU power and parallel concurrency, each agent runs on a separate core as a single process.

The agents have two basic functionalities: local computation and global communication. The system schematic of the two functionalities can be seen in Figure 5.2. The local computation is based on the traffic models described in the linear model experiment and the nonlinear experiment. Because the traffic models are fixed mathematical models and the initial values are also fixed, the simulation results are exactly the same when the experiment is replicated. This is a deterministic experiment, NOT a random experiment. The global communication is done by sending data to its upstream and downstream agents. From the graphical representation of the ring road networks, each subnetwork has a number of upstream and downstream subnetworks. Each agent has a upstream list and a downstream list with its upstream and downstream agents respectively. Each agent communicates the supply to its upstream agents and the demand to its downstream agents when the local computation is finished. The format of the communicated data is an array of three real numbers which represent the demand or the supply of the next 30 minutes.

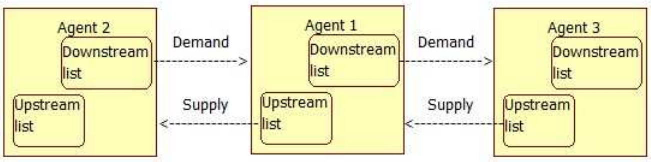


Figure 5.2: System schematic

The functionality of the global communication is the same for all agents, however the functionality of the local computation is different for different agents since the number of segments and the length of the segments are different for each subnetwork. The core agent source code can be seen in B.4.

5.5.2 Distributed prediction of a linear model

In this chapter, a simple ring network with subnetwork 1 and 2 as shown in Fig. 5.1. The subnetwork 1 consists of four sections or cells with one on-ramp and one off-ramp. The subnetwork 2 consists of five cells with one on-ramp and two off-ramps. In this case study, a linear model is considered for this simple ring network.

Linear model

The flow of cells in Subnetwork 1 is modeled as:

$$x_1(t+1) = a_{11}x_1(t) + b_1 u_{11}(t) \tag{5.20}$$

$$x_2(t+1) = a_{21}x_1(t) + a_{22}x_2(t) - h_2 x_2(t) \tag{5.21}$$

$$x_3(t+1) = a_{32}x_2(t) + a_{33}x_3(t) + m_3 a_{31}(t) \tag{5.22}$$

$$x_4(t+1) = a_{43}x_3(t) + a_{44}x_4(t) - h_4 x_4(t) \tag{5.23}$$

$$y_1(t+1) = h_4 x_4(t) \tag{5.24}$$

The equations are then collected in the following system for all sections.

$$x(t+1) = A_1 x(t) + B_1 u(t), \tag{5.25}$$

$$y_1(t+1) = C_1 x(t), \tag{5.26}$$

$$x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{pmatrix}, \quad u(t) = \begin{pmatrix} u_{11}(t) \\ a_{31}(t) \end{pmatrix}, \tag{5.27}$$

$$A_1 = \begin{pmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22}-h_2 & 0 & 0 \\ 0 & a_{32} & a_{33} & 0 \\ 0 & 0 & a_{43} & a_{44}-h4 \end{pmatrix}, \quad B_1 = \begin{pmatrix} b_1 & 0 \\ 0 & 0 \\ 0 & m_3 \\ 0 & 0 \end{pmatrix},$$

$$C_1 = \begin{pmatrix} 0 & 0 & 0 & h_4 \end{pmatrix}.$$

The traffic flow in Subnetwork 2 is modeled next. Again the state is denoted by the symbol x and the reader is warned not to identify this vector with that of Subnetwork

1.

$$\begin{aligned}
x_1(t+1) &= A_{11}x_1(t)+B_1u_{12}(t) && (5.28)\\
x_2(t+1) &= A_{21}x_1(t)+A_{22}x_2(t)-H_2x_2(t) && (5.29)\\
x_3(t+1) &= A_{32}x_2(t)+A_{33}x_3(t)+M_3a_{32}(t) && (5.30)\\
x_4(t+1) &= A_{43}x_3(t)+A_{44}x_4(t)-H_4x_4(t) && (5.31)\\
x_5(t+1) &= A_{54}x_4(t)+A_{55}x_5(t)-H_5x_5(t) && (5.32)\\
y_2(t+1) &= H_5x_5(t) && (5.33)
\end{aligned}$$

The equations are then collected in the following system for all sections.

$$\begin{aligned}
x(t+1) &= A_2x(t)+B_2u(t), && (5.34)\\
y(t+1) &= C_2x(t), && (5.35)\\
x(t) &= \begin{pmatrix} x_1(t)\\ x_2(t)\\ x_3(t)\\ x_4(t)\\ x_5(t)\end{pmatrix}, \ u(t)=\begin{pmatrix} u_{12}(t) & a_{32}(t)\end{pmatrix}, && (5.36)\\
A_2 &= \begin{pmatrix} A_{11} & 0 & 0 & 0 & 0\\ A_{21} & A_{22}-H_2 & 0 & 0 & 0\\ 0 & A_{32} & A_{33} & 0 & 0\\ 0 & 0 & A_{43} & A_{44}-H4 & 0\\ 0 & 0 & 0 & A_{54} & A_{55}-H5\end{pmatrix}, \ B_2=\begin{pmatrix} B_1 & 0\\ 0 & 0\\ 0 & M_3\\ 0 & 0\\ 0 & 0\end{pmatrix},\\
C_1 &= \begin{pmatrix} 0 & 0 & 0 & 0 & H_5\end{pmatrix}.
\end{aligned}$$

The interconnection of the two subnetworks is then described by the relations,

$$u_1(t)=y_1(t),\ \ u_2(t)=y_2(t). \qquad (5.37)$$

The parameter values of the two subnetworks were selected to be,

$$\begin{aligned}
0.5 &= a_{11}=a_{21}=a_{22}=a_{32}=a_{33}=a_{43}=a_{44},\\
0.5 &= A_{11}=A_{21}=A_{22}=A_{32}=A_{33}=A_{43}=A_{44}=A_{54}=A_{55},\\
0.1 &= h_2=h_4=H_2=H_4=H_5=m_3=M_3,\\
1 &= b_1=B_1=1.
\end{aligned}$$

The traffic inflow from the two on-ramps were selected to be constant with the values $a_3(t)=A_3(t)=1000$.

Random inflow initial guess

If the current traffic state is unknown, initial values of inflow in the subnetwork 1 ($u_{11}(t)$) and the subnetwork 2 ($u_{12}(t)$) are randomly selected. In this chapter, the constant inflow $u_{11}(t)=1000$ and $u_{12}(t)=1000$ are selected. Six iterations are needed to have a consistent simulation. The results of the six iterations can be read in Table 5.1.

Table 5.1: Six iterations of two subnetworks

1st iteration	Input	Output	2nd iteration	Input	Output
Subnetwork 1	1000	500	Subnetwork 1	400	230
Subnetwork 2	1000	400	Subnetwork 2	500	220
3rd iteration	Input	Output	4th iteration	Input	Output
Subnetwork 1	220	150	Subnetwork 1	125	110
Subnetwork 2	230	125	Subnetwork 2	150	90
5th iteration	Input	Output	6th iteration	Input	Output
Subnetwork 1	90	90	Subnetwork 1	80	90
Subnetwork 2	110	80	Subnetwork 2	90	80

We can conclude that iterations are needed if the initial guess of the inflow is far away from the convergence value.

Adaptive prediction as inflow initial guess

In case of on-line traffic prediction, we assume that the current traffic state is known. Based on the current traffic state, an adaptive filtering algorithm Wang et al. (2011a), Wang et al. (2014a) gives an accurate prediction of traffic demand on the boundary of a network. Using this as the initial input, the predicted inflow provides a rather close initial guess to the convergence value. In this case study, we assume that the result from adaptive filtering gives about 5% difference from the convergence value. Therefore, a constant inflow $u_{11}(t) = 76$ and $u_{12}(t) = 86$ are selected. Then, the results are almost the same as the 6th iteration in table 5.1. The predicted outputs of the two network parts achieve consistency in one iteration.

We can conclude that only one iteration is needed if the initial guess, by using adaptive filtering algorithm, is close to the convergence value.

Performance

Performance in terms of communication, computation and accuracy can be described as follows.

For one subnetwork, only one iteration is needed for the linear model. If the inflow on the boundary point is accurately predicted by the adaptive filter, iteration is not needed any more. The computation time and communication time have been measured and are shown in Table 5.2. Each communication between two agents take 0.0001 sec. The traditional parallel simulation synchronizes the inflow and the outflow at boundary points at each time step. If we are trying to simulate for a half hour and 5 sec for each time step, agents have to communicate 360 times at the boundary points. This means the communication time is 0.036 sec. The computation time of a centralized simulation is also measured and shown in Table 5.2.

Table 5.2: Comparison of computation time and communication time.

Simulation method	Computation sec	Communication sec
Traditional parallel	0.002	0.0360
Centralized	0.004	0
Proposed parallel	adaptive prediction initial guess 0.004 random initial guess 0.012	0.0002 0.0005

For the traditional parallel prediction approach, the computation of the fastest agent and the slowest agent take 0.0015 sec and 0.002 sec respectively. For the proposed parallel prediction approach, the computation time is two times that of the traditional parallel prediction approach in case of using adaptive prediction as initial guess; the computation time is six times that of the traditional parallel prediction approach in case of a random initial guess.

In general, the traditional parallel simulation takes 0.038 sec, since it depends on the simulation time of the slowest agent (0.002 sec) and the communication time (0.036 sec). However, the proposed parallel simulation takes 0.0042 sec in case of adaptive prediction and 0.0125 sec in the worst case (random initial guess). On average, the proposed parallel simulation is much faster than the traditional parallel simulation. However, for the small toy network, the proposed parallel simulation does not show any advantages in terms of the total time spent comparing with a centralized simulation. The advantage of the proposed approach will be demonstrated in a larger road network case study later in this chapter.

Considering accuracy of the parallel simulation, we can observe the following. By using the iterative approach, we can guarantee that the inflow of the downstream subnetwork exactly equals the outflow of the upstream subnetwork. So the accuracy of the proposed parallel simulation is exactly the same as the other two approaches.

5.5.3 Distributed prediction of a nonlinear model

In this case study, the road network in the Amsterdam area (see Figure 5.3) including A10, A4, A1, A8 and A2 is considered.

Network subdivision

The subdivision procedure formulated above is now illustrated for the Amsterdam network.

Example 5.5.1 *Consider the* Motorway Ring of Amsterdam *consisting of only the ring and its nodes with the four motorways ending on it, the A1, A2, A4, and A8, and the*

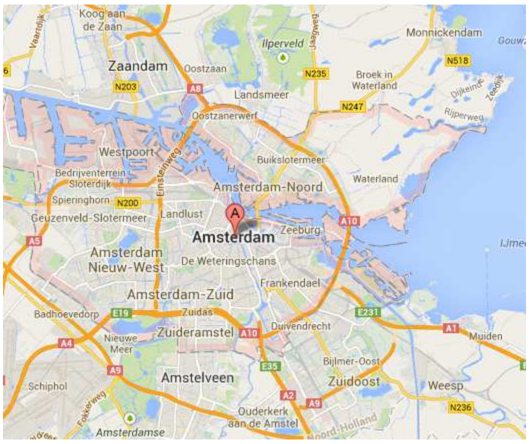


Figure 5.3: Amsterdam Ringnetwork A.10. (Photocredit Google maps).

Table 5.3: Subnetwork mode.

444	193	500	309	500	364	356	500	500	500	439	438	500
-1	+1	0	0	0	-1	+1	0	0	0	-1	+1	0

on-ramps and off-ramps of this ring. One can distinguish in a ring the clock-wise *and the* counter clock-wise traffic flows.

Partition the ring into four subnetworks each of which starts at a junction with one of the four motorways and ends at the next intersection or junction. The subnetworks are then labeled by the tuple of start and end points in terms of the motorways connected to those points. Due to the considered clock-wise traffic flow, the subnetworks are labeled as (A1,A2), (A2,A4), (A4,A8), and (A8,A1). The stretch (A8,A1) is long and may be partitioned into two consecutive pieces if this is regarded as appropriate.

See Fig. 5.3 for the map of the Amsterdam Ring.

The Network is decomposed into 16 subnetworks (one directional road) and further to cells from a shape file automatically. Figure 5.4 shows the subnetworks by using different colors. The basic steps are as follows:

1. Create major junctions in a semi-automatic way Wang et al. (2009b,c).

2. A subnetwork can be generated automatically in the following way: Find all the links from a split point of a junction to a merge point of the neighboring junction by using the shortest path algorithm.

3. Links are further subdivided into cells: the maximal length of a cell is 500 meters.

The number of cells for the subnetworks are between 13 and 33. One of the example the subnetwork model is shown in table 5.3. The first row is the cells with length in meters; the second row is the on-ramp (1) and the off-ramp (-1).

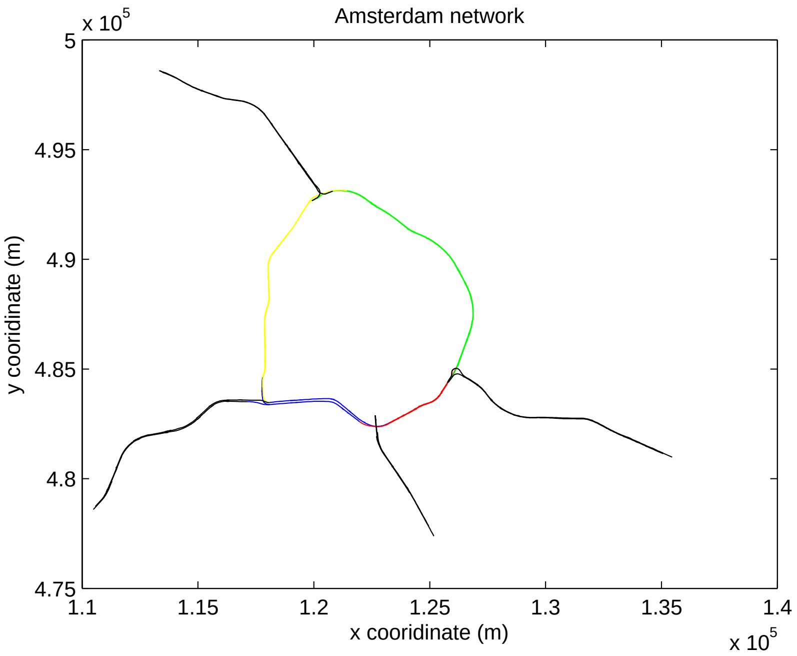


Figure 5.4: Amsterdam network.

Nonlinear model

The FastLane model has been implemented in Matlab since 2008 by the Department of Civil Engineering of the Delft University of Technology (TUD). Fastlane is a multi-class first-order traffic flow model van Lint et al. (2008). The Fastlane model considers model equations for different user classes such as passenger cars and trucks. For each user class, the discrete conservation equation (5.38) holds.

$$\rho_{u,i}(t+\Delta t) = \rho_{u,i}(t) + \frac{\Delta t}{\Delta x}(q_{u,i+1/2}(t) - q_{u,i-1/2}(t)) \tag{5.38}$$

Where Δt denotes the discrete time step which satisfies $\Delta t \leq \frac{\Delta x}{\upsilon_{max}}$

Each user class has its own fundamental diagram equation (5.39),

$$q_u = Q_u(\rho). \tag{5.39}$$

For each user class, the relation of the traffic flow (5.40) holds.

$$q_u = \rho_u \upsilon_u \tag{5.40}$$

Where ρ_u denotes the class specific density; q_u denotes the class specific flow; v_u denotes the class specific velocity.

Based on the equations (5.38,5.39,5.40), one is able to calculate ρ_u, q_u and v_u. The equation (5.41) is the total density which equals a weighted summation over all class specific densities.

$$\rho = \sum_u \eta_u \rho_u \tag{5.41}$$

Table 5.4: 1st iteration of four inner ring subnetworks

1st iteration	Input	Output
Subnetwork 1	500	635
Subnetwork 2	500	250
Subnetwork 3	500	737
Subnetwork 4	500	823

Where η_u denotes the weight of specific user class.

In this case study, a single class FastLane model has been implemented on top of AgentScape for the A10 ring network (see figure 5.3).

Subdivision is described in the previous subsection. For each subnetwork, one agent is created in the AgentScape. The subnetwork mode is read into the traffic model to perform a simulation of the subnetwork. 16 subnetworks are simulated in parallel.

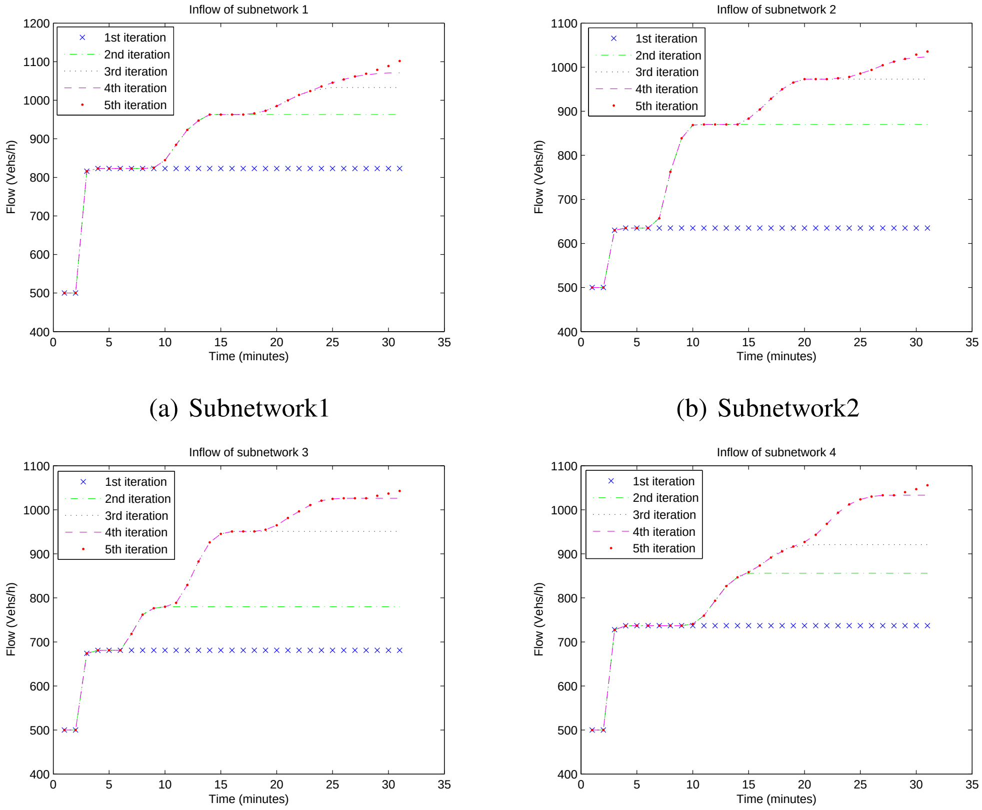
Random inflow initial guess

If the current traffic state is unknown, initial values of inflow in the 16 subnetworks are randomly selected. In this chapter, the initial inflow of 500 veh/h from motorways, the constant inflow of 100 veh/h from on-ramps and the constant turn fraction 0.1 are selected for each subnetwork. In case of free flow, it can be treated as a linear model which is described above. Five iterations are needed to achieve consistency of predictions in the 16 subnetworks. The subnetworks on A1, A2, A4, and A8 only need one iteration. The subnetworks on the ring road need a few more iterations. The results of the first iteration for the four inner ring subnetworks can be read in Table 5.4. Subnetwork 1, 2, 3 and 4 are the four subnetworks of the A10 inner ring.

Inflow and outflow of each subnetwork can be seen from Figure 5.5 and Figure 5.6 respectively. We can see from the figures that inflow and outflow are updated at each iteration. On the other hand, the results prove the recursive method.

In case of congested traffic, the fixpoint method will be used. We assume that subnetwork 2 is congested such that it has a fixed outflow (250 veh/h). Five iterations are needed to achieve consistency at the boundaries.

Inflow and outflow of each subnetwork can be seen from Figure 5.7 and Figure 5.8 respectively. We can see from the figures that inflow and outflow are corrected at each iteration. Although four iterations are needed in total, not all subnetworks have to iterate four times. The subnetworks on A1, A2, A4, and A8 only need one iteration and some subnetworks need 2 or 3 iterations.



(a) Subnetwork1

(b) Subnetwork2

(c) Subnetwork3

(d) Subnetwork4

Figure 5.5: Inflow of each subnetwork

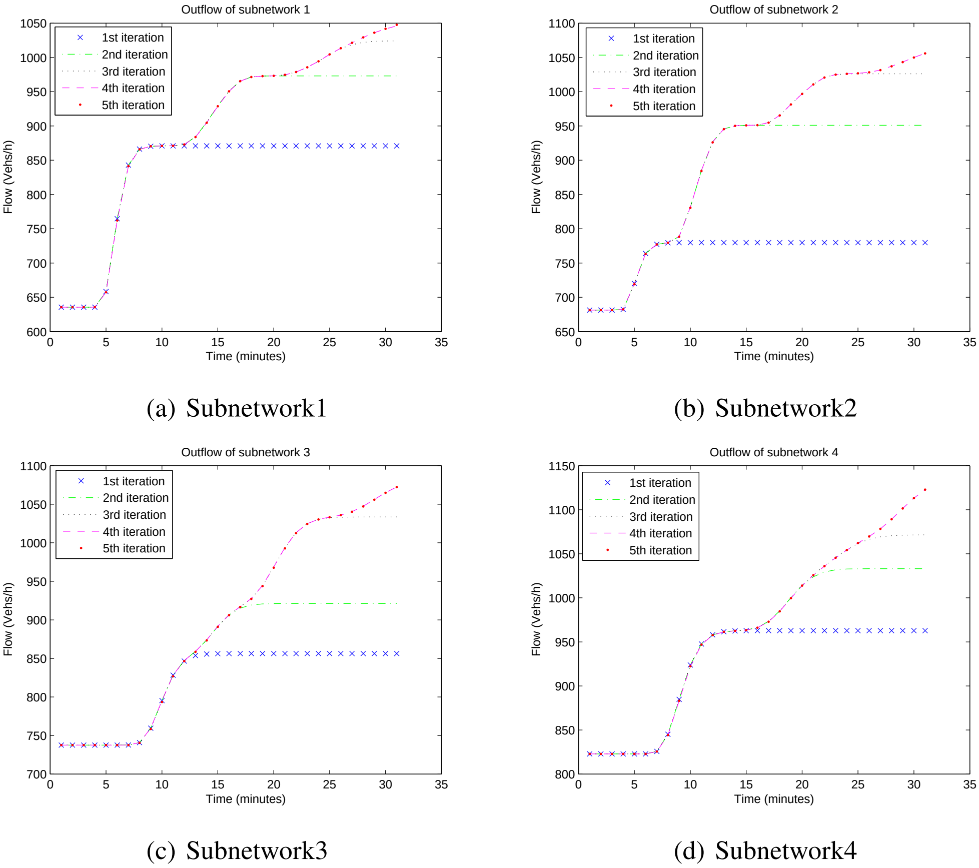
Adaptive prediction as inflow initial guess

In case of on-line traffic prediction, we assume that the current traffic state is known. Based on the current traffic state, an adaptive filtering algorithm Wang et al. (2011a), Wang et al. (2014a) gives an accurate prediction of traffic demand on the boundary of a network. Using this as the initial input, the predicted inflow provides a rather close initial guess to the convergence value. After the network is simulated to a steady state, we assume that the inflows on the boundary of A8 increase 10 percent. We can see the simulation result from figure 5.9. The conclusion is the same as in the linear model, that only one iteration is needed if the initial guess is close to the convergence value.

Performance

Performance in terms of communication, computation and accuracy is discussed in this subsection.

The computation time and the communication time are discussed and measured for the nonlinear model for both free flow and congested traffic. If the inflow on the boundary point is accurately predicted by the adaptive filter, only one iteration is needed. The computation time and communication time are measured and shown below. Each


Outflow of subnetwork 1

Outflow of subnetwork 2

Outflow of subnetwork 3

Outflow of subnetwork 4

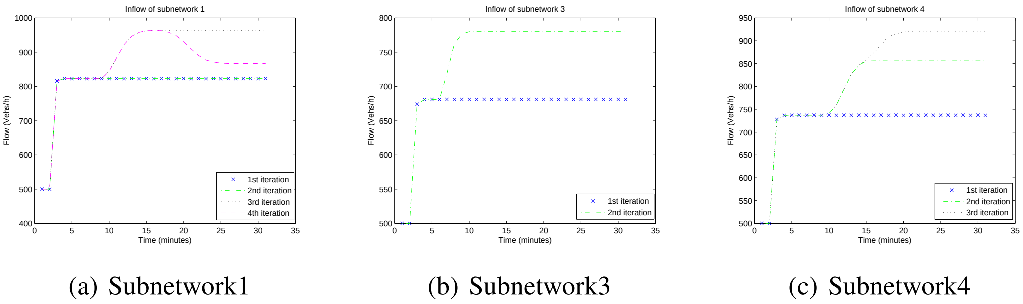
(a) Subnetwork1

(b) Subnetwork2

(c) Subnetwork3

(d) Subnetwork4

Figure 5.6: Outflow of each subnetwork


Inflow of subnetwork 1

Inflow of subnetwork 3

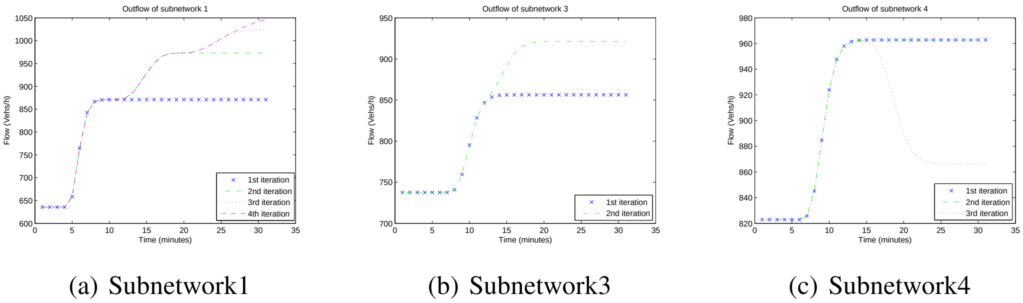
Inflow of subnetwork 4

(a) Subnetwork1

(b) Subnetwork3

(c) Subnetwork4

Figure 5.7: Inflow of each subnetwork


Outflow of subnetwork 1

Outflow of subnetwork 3

Outflow of subnetwork 4

(a) Subnetwork1

(b) Subnetwork3

(c) Subnetwork4

Figure 5.8: Outflow of each subnetwork

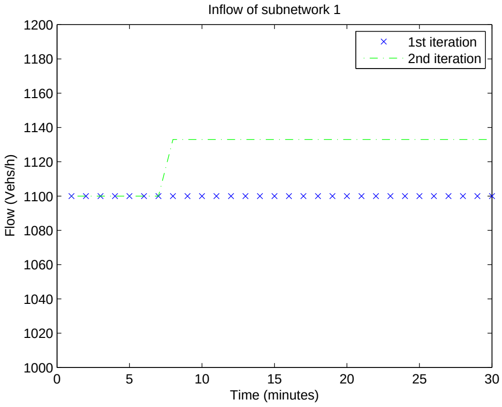


Figure 5.9: Inflow of subnetwork 1.

Table 5.5: Comparison of computation time and communication time.

Simulation method	Computation sec	Communication sec
Traditional parallel	0.015	0.0360
Centralized	0.078	0
Proposed parallel	adaptive prediction initial guess 0.030 random initial guess 0.075	0.0002 0.0005

instance of communication between two agents takes 0.0001 sec. The traditional parallel simulation synchronizes the inflow and the outflow at boundary points at each time step. If we are trying to simulate for a half hour, using 5 sec for each time step, agents have to communicate 360 times at the boundary points. This means the communication time is 0.036 sec. The computation time of a centralized simulation is also measured and shown in Table 5.5.

For the traditional parallel prediction approach, the computation of the fastest agent (For the subnetwork with 13 cells) and the slowest agent (for the subnetwork with 33 cells) take 0.002 sec and 0.015 sec respectively. For the proposed parallel prediction approach, the computation time is the same as the traditional parallel prediction approach in case of the adaptive filtering algorithm as initial guess; the computation time is five times the traditional parallel prediction approach in case of a random initial guess (since five iterations are needed).

In general, the traditional parallel simulation takes 0.051 sec, since it depends on the simulation time of the slowest agent (0.015 sec) and the communication time (0.036 sec). However, the proposed parallel simulation takes 0.0151 sec in the best case and 0.0751 sec in the best case. Comparison with a centralized simulation shows that it is lower than the proposed parallel simulation. In addition, the centralized simulation is not scalable. If the size of the network further increases, the proposed approach will have increasing advantages in terms of speed. The proposed parallel simulation does not show any benefits compared with the traditional parallel simulation in terms of total speed, using a random initial guess, but the proposed parallel simulation with an adaptive filtering algorithm for the initial guess is faster than the traditional parallel simulation.

Accuracy of the proposed parallel simulation compared with both traditional parallel simulation and centralized simulation is analyzed. Due to the congested subnetwork, the outflow of the subnetwork can be considered as a fixed outflow such that we can guarantee that the inflow of the downstream subnetwork exactly equals the outflow of the upstream subnetwork. The accuracy of the proposed parallel simulation is exactly the same as for the other two approaches.

5.6 Conclusions

This chapter presents a scalable algorithm for the prediction of traffic flow in a large-scale motorway network. Both a linear and a nonlinear traffic model are demonstrated for the proposed algorithm. It is illustrated by applying the algorithm to a simple toy network and the Amsterdam ring network. In case of both the linear and nonlinear model, the fixpoint method with adaptive filtering algorithm as an initial guess, is applied to predict a large-scale motorway network with two iterations. We can conclude that the total computation time of the proposed algorithm is much shorter than that of the traditional parallel prediction algorithm and the centralized prediction algorithm. The most important feature of the proposed algorithm for both linear and nonlinear cases is that it benefits from the asynchronous iterative approach. This is a scalable approach for large-scale motorway networks, even in case of loss of communication. The computer program is considered for implementation into the software of traffic control centers in The Netherlands.

Chapter 6

Multilevel Network Management: the QHM Framework and the SCM Operational System[1]

6.1 Introduction

The current status of NM is that there are a number of systems installed at many places worldwide, that do this kind of coordination of local measures, but their scope is still limited. Either the area is limited, or the coordination applies only to one kind of local measure. There are many systems for the coordination of traffic signals in urban areas Hamilton et al. (2013), Pooran & Martinez (2011), Katwijk (2008). And there are systems for the coordination of ramp metering systems at the boundary of urban and motorway networks Papamichail et al. (2010b). The SCM (Scenario Coordination Module, Wang et al. (2010b)) in Amsterdam, operational from September 2010 till end of 2014, was among the very few systems in the world that covered a wider area (the city of Amsterdam and the motorways surrounding it), that covered different types of roads (motorways and urban roads) and that covered a variety of traffic management measures: traffic signals, ramp metering systems and variable message signs.

What is most needed in order to progress beyond the current status, is a framework or system architecture for NM that covers the different aspects of the problem in a coherent way, most notably the IT-technical aspect of connecting systems and the traffic engineering aspect of how to do NM. On the one hand, the framework must serve the purpose of putting together all the components needed for NM. Most components are existing traffic management systems, the majority working locally, and a variety of existing sensors and actuators. On the other hand, the framework must offer a scalable approach to the operations of NM. This approach is the main topic of this chapter. The connectivity has been treated in other publications Vrancken et al. (2012). These

[1] This chapter is based on my publication Vrancken et al. (2013)

two aspects have a relationship, as the approach to NM strongly influences the way the IT-technical connections between the NM-system and the legacy components are made.

The challenge of NM is also related to the other big challenge in road traffic, the emergence of cooperative systems (CS), which are information processing systems for a variety of purposes, that consist of components within vehicles, along the roadside and in various kinds of back offices and traffic management centres Sotelo et al. (2012). The idea that car-to-car systems will make traditional traffic management by authorities superfluous, has been countered often enough Hoogendoorn et al. (2011). But CS offer various functions that will be essential for fully fledged NM. For instance, the equipment within vehicles can serve as ubiquitous sensors and ubiquitous actuators. In this way, the two main developments in road traffic, NM and CS are strongly related.

The framework QHM (Quantitative Hierarchical Model) considers Network Management of road traffic first of all as a problem of complexity and scalability. A single controller for one crossing is doable, although this is already a complex problem. Scaling this control problem to the level of a whole country, and still using one control instance is totally impossible, because of the complexity of this problem. It is inevitable to have many control instances, and to coordinate them (which is synonymous with controlling them). For a large network, it will even be necessary to have many coordinators, that themselves also need coordination, resulting in a multi-level control system. From a complexity point of view, all of these control instances must be able to shield off much of their internal complexity, such that the complexity as seen by its coordinator is far less. QHM is about how to organize this coordination of many control instances, and thereby scale up traffic management from the very local to networks of any size. The two key ingredients are multi-level coordination and complexity shielding.

There are two sources of inspiration. One is the governance of countries. In virtually all countries, government is hierarchically organized: a country is split-up into states or provinces, with each lower level unit having its own government. Such a unit is still too big for one governing instance. Therefore it is split up further, into districts, counties, municipalities, etc. This is a recursive, hierarchical split-up, needed to make the governance of a country doable by human beings, of whom it is known that each individual can handle only very little complexity.

The second source of inspiration is Systems Engineering (SE) Kossiakoff et al. (2011), which specializes in the design and management of large, often multi disciplinary, systems. It turns out that a number of powerful concepts from SE are applicable to a very broad range of systems, such as air- and spacecraft, the human body or large factories. One principle we have already mentioned: hierarchical decomposition. The notion of 'system' is essentially hierarchical: a system contains subsystems, that are themselves fully fledged systems, also containing subsystems. The second concept is a system's boundary. A system lives within an environment and it has a boundary. The boundary is the interface where the interactions take place between system and environment. Whatever happens inside a system is irrelevant for the environment if it has no effect on the boundary. And the other way around: whatever happens in the environment is irrelevant for the internals of a system if it has no effect on the boundary. The boundary results in strong complexity shielding in both directions. This can be taken even further by making boundary agreements between a system and its environment, such that the boundary agreements change on a time scale larger than the time scale relevant for internal management. It this is possible, the internal management of a system is effectively decoupled from its environment.

Without measures to reduce complexity effectively, a system's complexity grows exponentially with its size. If it is possible to apply abstraction to boundary behavior of its subsystems, and to make agreements on boundary behavior on a large enough time scale, then complexity becomes linear in the size of a system, which is the best one can hope for.

QHM is essentially about applying these concepts to road networks and the problem of network management, using a hierarchical decomposition of the road network, strong abstraction of traffic behavior to network boundaries, network configuration with boundary agreements and multi-level control of traffic.

6.2 Problem Statement

The motivation for Network Management stems first of all from the observation that many traffic problems, most notably congestion, are network-related. Without the network context, solving congestion at one place often means creating it at another place. A more practical motivation stems from the management of the Dutch freeways. There are five regional control centres in The Netherlands and one national centre, that continuously manage traffic on the freeways, with measures such as ramp metering, dynamic speed control, routing advice, variable message signs, congestion warning, lane closures, and hard shoulder lanes. Yet this management is still not entirely satisfactory, given the frequent congestion that still occurs, especially in urban areas with strong exchange of traffic between freeways and the urban network.

As described in the previous section, the problem has to be split up into smaller parts, each with its own management, shielding off to a large extent the internal complexity, such that only little complexity remains at the top level. This can be illustrated with the control problem of a car. To control the driving behavior of the whole car, one can split it up into the main components driver, engine, transmission and wheels. Each of these components has its own internal control and can be abstracted for overall control to reduce its complexity: the driver is reduced to steering wheel and pedal movements, the engine is reduced to rotational speed and torque, the transmission is reduced to a ratio between input and output rotational speed and the wheels are reduced to rotational speed and direction. The overall control can now make sure that changes in rotational speed of the different components take place smoothly, that speed is not too high when the car turns right or left, etc. In this case, a multi-level control system is needed, but the components at the lower level are quite different, so their control is also quite different, which is not a big problem as there are only few of them. In the case of Network Management of large networks this is not enough. In this case we must expect that the hierarchical decomposition results in huge numbers of components. Then it is necessary to obtain a fair degree of uniformity in the control of all of these components, otherwise scalability would be defeated by the sheer amount of work to develop and maintain all of these different control systems.

Problem 6.2.1 *Consider a large-scale road network. Produce a multilevel control system for such a network. Synthesize a set of largely uniform control laws for all subsystems at all levels of the multilevel control system such that the closed-loop multilevel control system satisfies Key Performance Indicator according to traffic policy.*

6.3 Input Data, Actuators and Control Objectives for Network Management

A control system needs input data by monitoring its target system, it needs actuators to exert influence on its target system, and it needs control objectives, stemming from the desires of the owner of the system, in order to know what to do. In this section, we will describe these three aspects for Network Management, focussing on a coordinating control system, not at the lowest level of the multi-level control system. Such a coordinator has a geographical scope, the area it is responsible for, which we will call network N. For input data and actuators, it is also important to be able to abstract from the physical equipment. To that end, we consider data and actuator effects on the boundary of N.

6.3.1 Input Data

Typical input data for boundary points are traffic properties such as speed, flow, the OD demand matrix (origins and destinations being the entries and exits of network N), the OD travel times matrix, and detection of various types of vehicles and vehicle properties. With today's means, all of these are more or less measurable. For instance, Bluetooth detection offers a cheap way to measure demand and travel time matrices (see chapter 8). In the coming era of cooperative systems, the phenomenon of FCD (Floating Car Data), together with a host of data fusion and data filtering techniques, will only further increase the already overwhelming amount of data that is available about traffic. The input data aspect does not seem to pose any serious limitations for Network Management.

6.3.2 Actuators

With actuators, one would like to influence traffic properties in boundary points such as, again, flow, speed, partial flows towards the various destinations, route choice, and limited or prioritized access for certain types of vehicles, such as trucks, public transport, etc. Influencing maximum speed is well-developed and relatively easy, both for urban and freeway traffic. Flow is harder. In urban traffic, flows can be effectively controlled, using cycle times of traffic signals. Signaling equipment with the one-vehicle-per-green property can even do this, to some extent, without stopping traffic. It will be necessary to control flow on freeways as well, even though this is not yet common today, because otherwise congestion will be hard to prevent and will hamper flow more than any flow control device ever will. Cooperative systems offer far better capabilities to control all of the basic traffic properties: both speed and density (intercar distance) can then be controlled from outside. The issue of societal acceptability is outside of the scope of this thesis. We can only say that, in the coming era of automated driving, with car sharing replacing more and more private car ownership, this issue will be radically different from what it is today. Controlling partial flows will be a challenge, even for cooperative systems. This will be possible only to a limited degree, at entry points where there is more than one lane. Route choice is different because it is upto the individual driver to choose its route. But still, information about travel times can influence route choice. Currently this is done to a limited extent with variable message signs. Cooperative systems will greatly improve the capabilities to influence route choice by offering far more detailed travel time information. The actual travel times on overly popular routes can be influenced by means of imposing speed limits. This makes sense by avoiding overloading of popular routes and stimulating better use of alternative routes, leading to better overall throughput.

6.3.3 Control Objectives

Control objectives are a complex aspect of Network Management. First of all, there is a variety of stakeholders in the traffic process. There are many drivers, with different preferences. Some favor low cost, some short travel times, some predictable travel times, some high driving comfort, some low emissions, etc. There are traffic operators, policy makers at various levels, inhabitants near heavily used roads, and so on. There are many objectives: safety, high throughput, absence of congestion, short travel times, low emissions, travel time reliability, driving comfort, and so on. Happily enough, these objectives are largely in line, especially in situations of underutilization. Safety is good for throughput, travel times, driving comfort. So does absence of congestion. In dense traffic however, the objectives are less in line. Then, for instance, a trade-off will have to be made between throughput, short travel times and reliable travel times. To complicate matters further, we observe that policy makers have a tendency to change their mind regularly and put more emphasis on some objectives rather than others. Here we focus primarily on the point of view of operators. They work for the collective optimum, which is derived from the traffic policy. If politics works well, the set of individual driver preferences and wishes from other stakeholders are expressed in the traffic policy. Most important for the operators is that they are able to translate policy preferences into measurable KPI's (Key Performance Indicators, such as TTP: Total Throughput, TTT: Total Travel Time, ADR: Average Delay Rate, and many others), that their systems can handle a variety of different KPI's, and that they know the effects of control actions on the KPI's.

6.4 Requirements for a Network Management Framework

Above we saw that the main problem of NM is the huge complexity of road networks and traffic. The main purpose of the framework is therefore to offer a way to unravel this complexity, such that each control instance in a network has only a limited amount of complexity to handle. But it has more purposes. For the practical implementability of NM, many connections have to be made. The framework must also offer a way to reduce the effort put into making and maintaining all these connections. Moreover, the goals of NM can vary over time, due to variations in traffic management policies. Network performance can be defined in different ways (amount of congestion, travel time reliability, total time spent in a network, etc.). At times safety or reduction of pollution may have priority over network performance. This means that the framework should be able to handle a variety of traffic policy goals. Finally, traffic behavior consists of a combination of regular, foreseeable patterns, unpredictable deviations from this regular pattern and disruptive, exceptional events, such as car accidents. The framework must cope with all three components of traffic behavior.

Currently, the foreseeable patterns are often expressed in so-called scenarios. These scenarios are important because the expected traffic patterns can only be formulated by humans, as the patterns are often influenced by events external to road traffic (weather, large sports events, road maintenance, ...). The scenarios constitute therefore an essentially human operator related aspect of traffic management.

The basic setting in which the requirements are formulated is that there are traffic management systems, either local or network-level, and that each system has the responsibility over a subnetwork (possible very small, f.i. just one crossing) of the total network of roads. This brings us to the following list of requirements.

- **Scope and scalability**:
  - The framework shall be applicable to arbitrarily large networks.
  - The framework shall be applicable to all kinds of networks (urban, motorway, interurban, border crossing, etc.).

  - The framework shall be applicable to all kinds of motorized traffic (person cars, motorcycles, trucks, public transport, etc.).

- **Connectivity**:
  - The framework shall facilitate the connection of NM systems with existing legacy sensors and actuators and local TM systems.
  - The framework shall allow connection of NM-systems to other NM-systems in neighboring areas, even if these neighboring systems are not framework-compliant.

- **Flexibility in Control**:
  - The framework shall support a variety of traffic management policies and a variety of combinations of network performance objectives.

- **Support for scenarios**:
  - The framework shall support the use of scenarios in traffic management.

6.5 The QHM framework

The description of QHM below covers the four main components of the framework:

- The recursive split up of networks.
- General properties of networks and traffic.
- General control principles.
- The multi-level control synthesis (network configuration + multi-agent control)

To illustrate the capabilities of the framework, two of its applications will be briefly described and we will show how the framework offers a new perspective on the problem of optimal traffic spreading and rerouting.

6.5.1 The Recursive Split-up of networks

When applying QHM to road traffic, the hierarchical decomposition or split up of the network can be made geographically: a network of roads can be split up into smaller networks. Every current approach to network management, to the best of our knowledge, does this Hoogendoorn et al. (2011); Kuhn et al. (2012); Vreeswijk & Blokpoel (2012); Immers et al. (2008b); Wang et al. (2012b). But what is not so common is to do it recursively: subnetworks are further split up into smaller subnetworks, just as in the governance of countries. Every network has a boundary: a finite set of points (cross sections through roads, in one driving direction). The boundary points of a network N are either entry points or exit points of N. Indeed, boundaries function as interfaces between N and its environment. Seen from outside, the behavior of a network N can be abstracted to the behavior of traffic on its boundary. The same holds the other way around: seen from inside N, the environment can be abstracted to traffic behaviour on this same boundary. We call this phenomenon the Modularity Principle. It gives a tremendous reduction in complexity. The geographic split up is so effective because of this strong reduction. But it doesn't need to be the only way of splitting up the traffic management problem. Just as in the governance of countries, one can also apply a functional split up, in addition to the geographic split up. For traffic, the functions could be: flow control, speed control, rerouting advice, parking control, safety control, fee collection, law enforcement, etc. Here, we limit ourselves mainly to the description of flow and speed control.

The recursive split up results in a tree structure of networks (Figure 6.1). Each network in the tree has its own control instance (in the sequel a network will often be identified with this control instance). For scalability, it is essential that all of these controllers are to a large extent uniform. Otherwise the amount of work for the development and maintenance of these controllers would still be far too much to make network management feasible. Below we will show that the description of the control procedure applies to both very large and very small networks and anything in between.

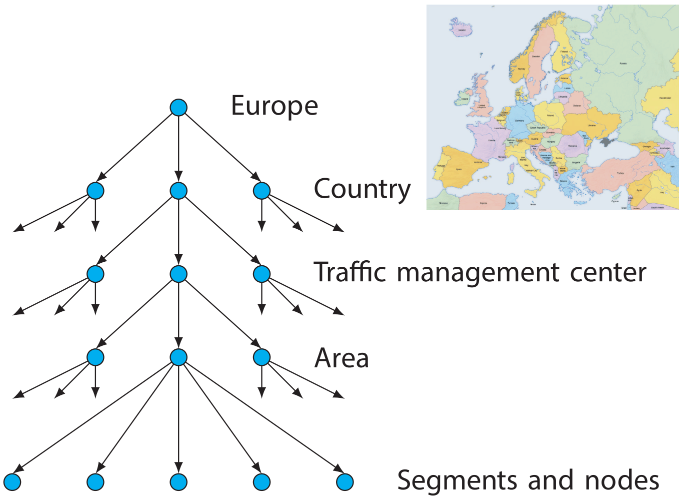


Figure 6.1: The recursive split-up of a large network.

For arbitrarily large networks, the split-up must be recursive and allow any number of levels. Just one level would not be sufficient: if the subnetworks of a given large network N are controllable without further split up, they must be very small, so there must be very many of them, which would make the control of N too complex. If the number of subnetworks is kept limited, they will be too big to allow control without further split up. The recursive way is the common and, to the best of our knowledge, the only way out.

Some terminology. All the subnetworks in the tree below a given network N are called the children of N. The direct subnetworks are called the direct children. N is called the parent of its children and direct parent of its direct children.

The recursive way introduces some complications that have to be addressed. Most notably, a boundary point can be entry point (or exit point) of several networks in the tree, at different levels, a so called *multi-border point*. Each of these networks exert some form of control on this point, and these control actions have to be consistent, a problem that will be addressed below. A given boundary point is exit point of a stack of networks and entry point of a stack of neighboring networks. The two stacks have a lowest common parent, which will play a role in the multi-level control synthesis below. It is obviously possible to break down a network into subnetworks, down to the level of segments and nodes, but to do it in some way optimally, for the problem of traffic control, turns out to be a very hard problem, outside of the scope of this framework overview. We limit ourselves to giving a few requirements for network split-ups. More detailed requirements can be found in chapter 5.

- Boundary points are under the influence of various neighboring networks. Therefore, they should be chosen in quiet places, where traffic control is easy, preferably in the middle of a segment, certainly not in the middle of a busy crossing.
- The traffic control problem is easier, as we will see below, when networks are closed under 'shortest route', which means that the shortest route between any two points in a network N falls within N. If this cannot be achieved fully, partially realizing this property helps.
- Any pair of adjacent networks A and B with a common direct parent, should be different in so-called child priority (to be explained below), which means that if congestion is unavoidable, then it should be clear where to store traffic first, in A or in B.

6.5.2 General Properties of Networks and Traffic

In order to formulate control principles and the multilevel control synthesis, a number of properties of networks and traffic are needed, which are described in this section.

Network Properties

Due to the recursive split-up, in QHM a network is not seen as a network of roads but as a network of networks. This is essential for scalability, as a road is not a scalable notion: bigger networks have more roads and this would soon become too complex. Networks, however, can be big or small, so a given network N can always be seen as a network of a small number of direct children. The size of the children is of no concern for N, as N only handles their boundary behavior. For internal control, each of the children can be split up further.

Basic Components. At the lowest level in a split-up, the basic components of networks are the segment (a stretch of road in one driving direction, without nodes), the choice point (a node where a segment splits up into two or more segments) and a merge point (a node where two or more segments join). The concept of lane is not needed for Network Management, as the effects of lanes on traffic are mostly local. Networks can thus be built up from basic components and from the joining of two adjacent networks.

Generalized boundary points. Boundaries consist of a finite set of entry and exit points. For scalability, it is necessary to be able to group boundary points, otherwise, big networks would have too many boundary points, making control too complex. In fact, groups of boundary points (a *gp* or *generalized point*), especially points close to each other, can be treated in many ways similar to single boundary points: the flow of a gp is the sum of the flows in its points, a route towards or through a gp is a route towards or through one of its points, the speed in a gp is a range of speeds in its points with a minimum and maximum value, the travel time from a gp to a gp has a minimum and maximum travel time, etc. The maximum internal travel time within a gp gives an upper bound for these gp-gp travel times. In modeling a network, the gp concept allows a trade-off between higher accuracy (more, smaller gp's) and less complexity (fewer, bigger gp's). Travel times can be estimated more accurately with smaller gp's, but there are more of them.

Traffic Properties

For a given network N, one can distinguish between four types of traffic, depending on the position of origin and destination relative to N:

- OO traffic: origin outside of N and destination outside of N.
- OI traffic: origin outside of N, destination inside N.
- IO traffic: origin inside N, destination outside N.
- II traffic: origin and destination inside N.

In the sequel only OO traffic will be considered, just to simplify the description. The other types of traffic do not pose any fundamental problems. Smaller networks with serious traffic control issues, such as ring roads, often have only OO-traffic, but for bigger networks, all types of traffic have to be taken into consideration.

Three components of traffic

Traffic can be considered to consist of three components: a regular, expected pattern, such as the morning rush hour, or the traffic caused by a sports event, unpredictable variations on this regular pattern, and exceptional events, such as accidents. Each component of traffic will be treated differently in the description of traffic control in the sections below.

Traffic control will always start with a given traffic policy and an expectation of traffic in the near future (in the order of half an hour): the regular, expected pattern. This pattern can be derived from historic data, possibly improved by filtering techniques using fresh data, as described in chapter 4. The expected traffic can be expressed as traffic demand at boundary points for the networks in the tree (to be done consistently at multi-border points). It does not need to consist of fixed values but can consist of functions of time. The specific variations that actually take place on this pattern are not predictable, but the statistical variance, which has an effect on traffic control, can be considered as part of the expected pattern.

Exceptional events are hard to model, therefore this will be restricted to the most common example: a traffic accident. Usually, an accident blocks (part of) a road. Such an event can thus be modelled as a network changing event: a segment is temporarily not part of the network, or its capacity is reduced.

Properties of traffic in a point

From outside of a network N, traffic behavior in N can be abstracted to traffic behavior on the finite set of boundary points of N. Thus it is important to consider which properties traffic can have in points. These properties can be properties of traffic, such as flow, speed and density, or properties of individual vehicles, such as type of vehicle (person car, truck, public transport vehicle, etc.), type of cargo (hazardous or not), intended destination, intended route, etc. Flow can be further divided into partial flows to the various destinations. Although traffic behavior in points is far simpler than traffic behavior in a network, it can still be rather complex. Below we will observe that network management can be done by managing traffic in a finite number of points. Such control actions can apply to any of the properties mentioned above. They can prescribe a maximum speed, or a maximum flow, or a maximum partial flow, or forbid trucks with hazardous goods to pass the point, etc.

Traffic Policy

Besides an expectation, one can also assume that there is a traffic policy in force for a given network. The traffic policy consists of the political decisions on how to manage traffic. Typically, policies express the relative importance of safety versus throughput versus environmental damage reduction, they determine priorities for the various parts of the network, can express limits on travel times, priorities for public transport, preferred routes for freight traffic, etc. Traffic policies result in KPI's (Key Performance Indicators) with which the effectiveness of network management can be measured.

6.5.3 General Control Principles in QHM

The first principle, the Modularity Principle (MP) has already been described above: it states that, for the surrounding network, traffic behavior in a network N can be abstracted to traffic behavior on the boundary of N. It has huge consequences for traffic control: the internal control of N can be separated from the control of the environment of N. They only depend on each other by a mutual agreement on boundary behavior, as will be detailed in the next section. It also means that the development of network management can be modularized: new methods for traffic management can be applied freely inside a network, as long as the boundary agreement is maintained. Often the area to which such a new method can be applied, is limited. Deployment within the framework QHM is possible when the method can maintain the boundary agreement, and in this way, the method becomes scalable. MP has also important consequences for the treatment of sensors and actuators. Equipment is a huge source of complexity in traffic control. A typical ramp metering system, for instance, has hundreds of parameters. MP offers a way to shield the many details of a specific piece of equipment. A piece of equipment usually has a small scope in the network (a crossing for traffic signals, an on-ramp and a stretch of freeway for ramp metering systems, a short segment of road for an induction loop, etc.). As usual, one can make a distinction between the effects within its scope and on the boundary of this area. The latter is most relevant for network management. An actuator can be abstracted to its effects on the boundary of its scope. A sensor can be abstracted to the kind of information it offers on the boundary of its scope. In this way, MP offers a uniform way to model a huge variety of roadside equipment.

The sequel may convey the message that Hierarchical Control is a complex process, and there is no denying that it actually is. Network Management is complex, there is no easy way to keep a network well performing. But one should keep in mind that MP renders the problem strongly modularized and that one can go gradually from (easy) robust performance (just making sure that a network stays fluid) to high performance. The example of TCP/IP in data communications shows that this learning process of gradual improvement can even be automated to some extent. So indeed, HC is complex, but the complexity is well modularized and therefore becomes manageable.

The second principle is called the Gating Principle (GP). It expresses that, in order to keep a network in good performing condition, the number of vehicles in the network must be controlled: on average, the number of vehicles allowed in should be equal to the number of vehicles allowed out. In that way, a well performing network can be expected to stay well performing. The principle is applied to all the networks in the tree, which means that one does not need to worry about the distribution of vehicles in a network: that is taken care of by the recursive application of the principle to all the children of a given network.

6.5.4 The Multi-level Control Synthesis

With the objectives, principles and trade-offs of the previous section, it is now possible to describe the process of control in more detail and to quantify it to some extent. The network control procedure below has two main components: how it can be built up recursively, from the control of basic components (segment, choice point, merge point), and how it handles each type of traffic (regular pattern, unpredictable variations, exceptional events). The recursive buildup shows how boundary agreements can be made between adjacent networks, how entry-exit priorities can be defined, how child priorities can be defined, and how control at multi-border points can be made consistent among the networks sharing such a boundary point.

Recursive buildup of the control procedure

For the recursive buildup of traffic control, one has to explain how the basic components are controlled, and how the control of the sum T of two adjacent networks A and B (see Figure 6.5) derives from the control of A and B. Well performing control of the latter two networks can be assumed as given, because of recursion.

The segment: The control of a segment (Figure 6.2) is mainly an application of GP and of the buffer space trade-off. Although this is the simplest control problem, at high demand it can already be a complex process. Typical control measures are: setting the maximum flow at the entry point (primarily dependent on the allowed outflow at b according to GP), setting max speed and giving information about expected travel time from a to b. Typical network configuration data from the segment are its maximum flow capacity, the speed capacity (the maximum sensible speed on the segment, independent of traffic conditions) and desired travel time, following from the traffic policy.

Figure 6.2: A road segment.

The choice point: The choice point (Figure 6.3) is the purest case of the rerouting problem. There may be too much traffic for one of the exits. Therefore, traffic at entry a must be limited destination specific: maximum partial flows have to be applied at a. Moreover, this case illustrates the notion of blocking congestion. Congestion at one of the exits may extend beyond the node, thereby blocking traffic for the other exit, which should be avoided. Part of the traffic may be diverted from one exit to the other, by giving information about travel times. High demand at, for instance, exit b may result, for some drivers, in the shortest travel time being via c, while at low demand it is via b. Typical control measures, in addition to those for segments, are setting maximum partial flow at a, per destination (b or c), and giving travel time information for a to b and a to c. Typical network configuration data, in addition to those for the segment, are the max partial flows, and the guaranteed travel times a to b and a to c, resulting from the traffic policy.

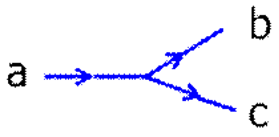


Figure 6.3: A Choice Point.

The merge point: The merge point (Figure 6.4) is the purest case in which a priority setting on inflows is needed. When there is more demand at a and b than allowed outflow at c (or when the inherent capacity of this tiny network cannot handle demand), a choice has to be made how much of traffic must be allowed in at each of the entries. This is not always a 50%-50% distribution, as congestion before one entry may have a more detrimental network effect than before the other entry. Therefore a priority is assigned to each of a and b, in the form of two positive numbers x, y with x+y = 1. When allowed outflow equals s, the allowed inflow is then (in principle) x*s, respectively y*s at the respective entries (in this example, we ignore the other factors that play a role in determining the maximum inflow at each entry). The merge point needs no extra control measures on top of what was mentioned for segment and choice point, but has a new contribution to the network configuration: the flow priority distribution for its entry points.

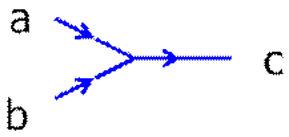


Figure 6.4: A Merge Point.

Joining two networks

The final step is about how to go from two arbitrary, adjacent networks A and B to the joined network T (Figure 6.5). Two children are enough to handle the general case: more children can be handled as several consecutive steps with two children.

All the entry points of a network are starting points of segments, therefore a network has no new control measures. The joining is entirely a matter of making boundary agreements, as it is only on the boundary that they have a relationship. If that works, then theoretically, the network management problem is solved for networks of arbitrary size. T has boundary agreements with its surrounding networks. A has a boundary agreement with B on their shared boundary. The boundary agreements must be such that they are suitable for internal traffic management, that they can be fulfilled and that they can be made recursively consistent at multiborder points. The nature of these boundary agreements can also be built up from basic components. The agreements are expressed in the typical properties of traffic and vehicles in a point. We limit ourselves to the key properties (flow, speed and entry-exit travel times) and give some examples (see Equations 6.1) of other kinds of properties that boundary agreements may deal with.

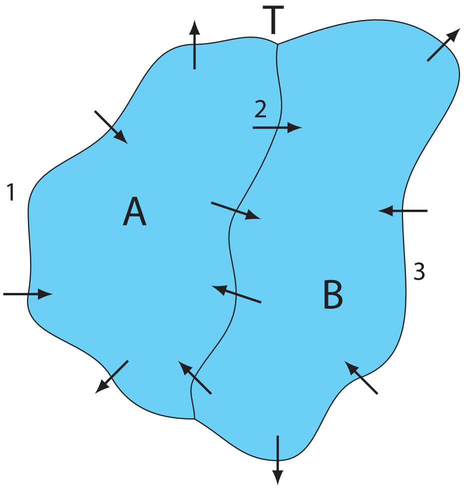


Figure 6.5: Parent-Child relationship between Priority Matrices.

$$
\begin{aligned}
T_{prio} &= \begin{pmatrix} T_{11}, T_{13} \\ T_{31}, T_{33} \end{pmatrix}, \; A_{prio} = \begin{pmatrix} A_{11}, A_{12} \\ A_{21}, A_{22} \end{pmatrix}, \; B_{prio} = \begin{pmatrix} B_{22}, B_{23} \\ B_{32}, B_{33} \end{pmatrix}, \\
T_{11} &= A_{11}, \; T_{33} = B_{33}, \; T_{13} = A_{12}B_{23}, \; T_{31} = B_{32}A_{21}. \qquad (6.1)
\end{aligned}
$$

A_{22} and B_{22} are (nearly) 0 because of (approximate) closedness under shortest path of the areas T, A and B.

Recursive Consistency of Flow Priorities

First we consider flow, as it is directly related to avoiding overloading of a network. The segment shows that one cannot make agreements about flows fixed for longer periods as the maximum inflow at a in Figure 6.2, depends on the outflow at b. So there is a general agreement that regularly new agreements have to be made on maximum inflows, with the time scale dependent on the time scale at which allowed outflows change, which, in a busy network, is a matter of seconds to minutes rather than hours. Moreover, a segment can make boundary agreements about other properties of traffic in points, such as speed, types of vehicles, maximum height of vehicles, types of cargo, etc.

The choice point adds to this that the maximum inflows have to be made specific per exit point. So the maximum inflow agreements are actually about maximum partial inflows towards the various exit points. The merge point adds that there must be a priority distribution among entry points. The combination of maximum partial flows per destination and priorities per entry point, lead, for a general network, to a flow priority matrix, with the rows corresponding to entries, and the columns to exits. This is illustrated in Figure 6.6 and Equation 6.2 for 2 entries and 2 exits. Maximum flows can still vary, it is just the ratio between maximum flows at the entry points that the entry flow priorities deal with. Therefore it is likely that such priority agreements can hold for longer time periods, in the order of half an hour to hours.

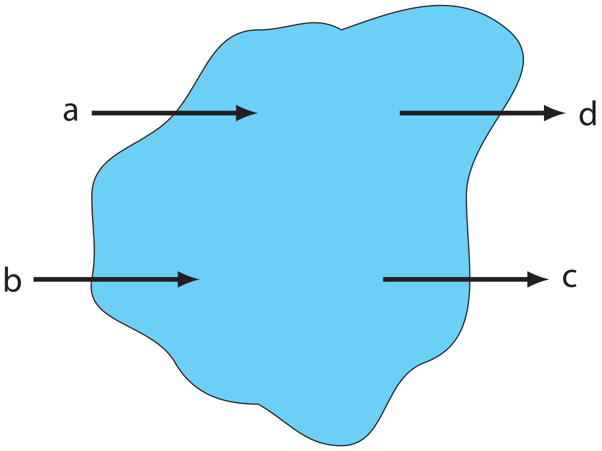


Figure 6.6: The Priority Matrix.

$$T_{prio} = \begin{pmatrix} p_{ac}, p_{ad} \\ p_{bc}, p_{bd} \end{pmatrix}, \; p_{ij} > 0, \; \sum_i p_{ij} = 1 \tag{6.2}$$

The key point now is to make these flow agreements recursively consistent at multi-border points. The main problem to do this for partial flows is that, for instance, network A in Figure 6.5 has different entries and exits than network T, so the partial flows for A are essentially different from those for T. Yet it is not hard to see that, from boundary 1 to boundary 3 for T, the flows consist of the sum of the different flows through boundary 2. For the priorities, this boils down to a matrix multiplication, as shown in equations 6.1. This shows that for flows and flow priorities, agreements can be made recursively consistent.

Speed is far easier, as the speed for the parent, at a multi-border entry point, is the same as the speed for the children. If networks at different levels have different requirements concerning the maximum speed, assuming that all of these requirements make sense, then one might agree on the minimum of all these maximum speeds. The same holds for restrictions on, for instance, hazardous goods: one can agree on the most restrictive agreement. For the control procedure as a whole, one must guarantee that networks impose only really necessary restrictions.

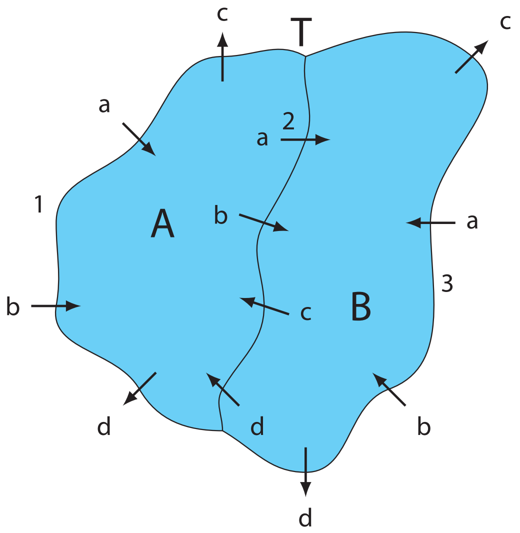


Figure 6.7: Parent-Child relationship between travel time matrices.

Recursive Consistency of Travel Times

For travel times, there is the same problem as for flow priorities: A (and B) have different entries and exits than T (see Figure 6.7). For entry p on border 1 and exit q on border 3, with tt_{pq} the travel time from p to q, we have $tt_{pq} = \min_{ronborder2}(tt_{pr} + tt_{rq})$. Again there is a relationship between the travel times matrix of T and those of A and B. In this case, it is about guaranteed travel times in a controlled area. Normally, travel times of two consecutive parts of a route cannot simply be added, because there is a time difference between arriving at the beginning of the first part and arriving at the beginning of the second part, but in the controlled area, the guaranteed travel times are more stable. They are part of the boundary agreements. Moreover, the one thing that matters for route choice is the relative order of travel times over the various routes, which is likely to be more stable than the travel times themselves. Figure 6.7 illustrates that the travel time from p to q on the border of T is the minimum of the times over the various routes passing through border 2. Here we assume, as usual, that T, A and B are closed under ”shortest path”, or that deviations from this property are negligible. This has as a consequence that border 2 is crossed only once and that all traveling from an entry of T to an exit of T takes place inside T.

6.5.5 Handling the different types of traffic

Now that it is clear how to make recursively consistent boundary agreements, we consider the question how to use these agreements for managing the different types of traffic distinguished in the section on Traffic Properties above: the regular pattern (the expected demand), unpredictable deviations from this pattern and exceptional, network changing events. The regular pattern will be translated into a network configuration, the unpredictable deviations will be handled by peer-to-peer requests between neighboring networks, also know as multi-agent control, and the network changing events will be handled by a network reconfiguration.

Network Configuration

A network configuration consists of the following components:

- A recursive split-up of the network under consideration (”the tree”)
- Flow priority matrices for all networks in the tree
- Guaranteed travel time matrices for all networks in the tree (if this is part of the traffic policy in force)
- Child priorities for each network in the tree
- Any other boundary agreements (speed limits, limitations for trucks, preferences for public transport, etc.)

Determining a network configuration starts first of all with a recursive split-up of the network, to be called *the tree*. The expected traffic demand can be expressed as the expected partial flows on the entry points towards the exit points of each network in the tree (we consider only OO-traffic). The demand must obey recursive consistency. According to this expected demand, flow priority matrices can be determined for all networks in the tree, as will be described below. Realistic flow priority matrices are based on historic data and are subject to gradual improvement. If the traffic policy includes guaranteed travel times, recursively consistent travel time matrices are determined for all networks in the tree, again based on historic data and subject to gradual improvement. Child priorities play a role in the multi-agent control described below. Finally, various other kinds of boundary agreements can be included. These are usually not subject to recursive consistency, so they are less problematic to determine.

Expected demand

The expected demand pattern determines several components of the network configuration, most notably the flow priority matrices. To determine such a configuration, which contains recursively consistent boundary agreements for all networks in the tree, one applies an iterative process, which will not be described in detail. Much of it is still subject of research. The idea is that agreements should on the one hand fit with the demand (in order to process as much of the demand as possible) and should allow good performance of a network. A parent may do a proposal to its direct children, using the formula in Figure 6.5. These children then determine if they can perform well under this boundary agreement. To that end, they have to consult their own direct children. And so on, till the bottom of the tree. The matrix formula 6.1 offers some freedom in priority setting per network. In the cases in which this process has been done by hand, it turned out not to be hard to find a workable solution. As we saw above, boundary agreements about other traffic properties may be part of the network configuration, but they are much easier to agree upon and to make recursively consistent. The same kind of process is applied in order to determine the matrices for guaranteed travel times.

The expected demand will most often be a function of time. For instance, the morning rush hour is not a constant flow, but gradually increases, reaches a top and then decreases to some mid-day value. This can be reflected in the network configuration, which may also be time dependent (expressed in time steps of, for instance, 10 minutes). Moreover, the expected demand for the next half hour can be improved using fresh data, with the help of the adaptive filtering techniques described in chapter 4.

Unpredictable deviations from the expected demand

Now we consider the unpredictable deviations from the expected pattern. As the configuration does not talk about maximum flows, this still has to be done. The allowed inflow is first of all determined by the allowed outflow (Gating Principle), which may vary on the time scale of seconds to minutes. Therefore, setting maximum inflows can best be handled by peer-to-peer interaction between neighboring networks, because of the short time scale on which this takes place. This works as follows. If a network experiences a problem, when traffic volume is approaching critical density and performance is about to deteriorate due to overloading, or when a network receives a maximum flow request on one or more of its exits, then it determines the maximum inflows that are necessary to stay well performing. For each entry point for which a request has to be issued, the highest network T in the stack of networks sharing this entry point, is determined. This network T, as we saw in section 6.5.1, has a neighbor N at this entry point, which is a direct child of the same parent. Then the request is sent from T to N. N, in its turn, translates the request on its exit point into consequences for its entry points and sends out the necessary requests in order to stay well performing.

An alternative way is to have the network that is the first to experience an imminent problem, to send maximum inflow requests to its neighbors directly and inform its parent about it. This is more complicated, but is also faster.

One request can lead to a cascade of requests throughout the network and appropriate measures should be taken to limit request storms. One way to do this is to limit the number of requests per unit of time and to apply a threshold below which requests do not lead to any cascading. For instance, changes in max flow of less than 10% may be handled purely locally. The further away from the original request point, the changes tend to become smaller and dampen out. Specifically for cyclic request storms, a child priority system (not to be confused with the flow priority system described above), is needed.

Preventing cyclic request chains: Child Priorities

A priority relation must be defined among the direct children of each network, with highest priority assigned to those networks that are most important for the functioning of the overall network. In determining the network split-up, it should be taken into account, according to the third requirement in section 6.5.1, that adjacent direct children of the same parent should have different child priority levels. This means that in case of imminent congestion, it is clear in which of two adjacent child networks, traffic can best be stored in order to limit the negative effects on overall network performance. How to determine effective child priorities is still subject of research, as is the relationship with the flow priorities. If the child priority relation is acyclic, and requests only go from high to low child priority, then cyclic request chains will be effectively prevented. An imminent overloading of a network is pushed back, according to the child priorities, to places where it does the least harm (congestion will occur mainly in low priority residential areas and parking lots). With the core network performing well, it may be expected that congestion in the low priority areas is less than in the current situation with congestion at arbitrary moments at arbitrary parts of the network, including high priority parts.

Exceptional events

Finally we have to handle exceptional events, which we limit to events that change the network, for instance accidents that block (partially) a segment or a node. Such events can occur all of a sudden, but will last easily for half an hour or longer. This means that the time scale points at a network reconfiguration, which seems to be the most appropriate way to handle this. Of course, some congestion cannot be avoided with this kind of unpredictable events. Also, the transition from one configuration to the next is a problem in this case, as this is not a gradual change. How to handle these transitions, is still subject of research.

6.5.6 Route Choice in QHM

Effective spreading of traffic can enhance the performance of a network. Drivers play an important role in traffic spreading as they have the freedom to choose their own route, and the collective effect of all of these route choices may be sub-optimal for network performance. Moreover, the problem is aggravated by an essential interdependence between route choice and traffic spreading. Route choice is based, to a large extent, on a perception of expected travel times, as most drivers tend to choose the time-minimal trip. Real travel times, however, depend on traffic density in the network, which in turn is strongly influenced by the collective route choices. QHM offers a way to escape from this vicious dependency.

In addition to this general problem of traffic assignment, the traffic control measure of rerouting is an important measure in daily practice of traffic management, applied in cases with either high unpredictability (for instance traffic accidents) or predictability (for instance road maintenance). In current practice, this is a difficult measure: there are many drivers involved at many places in the network, with a great variety of destinations, new routes must be calculated for many different groups of drivers, and this information must be communicated to the drivers, which, with the current means of a handful of visual roadside signs, is a challenge. Needless to say that the current solution is a matter of improvising with a very limited result, especially in the unpredictable case of traffic accidents. In the near future, when cooperative systems become more and more common, with extensive computation and communication facilities present in most vehicles, there are far more possibilities to address the problem of rerouting and traffic spreading effectively.

Routes and Route Choice in QHM

Routes can be viewed recursively. A route can be described as a series of networks that the route passes through. This can be done at any level of the tree, or even by combining different levels. In addition, one may add, per network, the entry and exit points (for simplicity's sake, we only consider OO-traffic). The travel time information measure mentioned above, means that vehicles, when approaching a network N, can ask network N for its Entry-Exit travel times matrix. In this way, shortest routes can be calculated and detailed/updated stepwise, as the driver passes through the route. In the area under QHM-control, travel times are much less dependent on collective route choice, as the guaranteed travel time matrix for each network is a boundary condition, part of the network configuration, and QHM-control strives at maintaining it by making sure that a network does not become overloaded.

The uncertainty that then remains is the waiting time before the area with guaranteed travel times can be entered. There will always be some uncertainty, as there is an inherent uncertainty in traffic demand. It is assumed, and this still awaits experimental verification, that if the core network performs well, then waiting times in the periphery will not be very long, and less than the waiting time in queues at arbitrary times and locations, which is now very common. The travel time information will cause an emergent effect towards an equilibrium in traffic spreading: highly popular routes will become less popular because of longer travel times (caused by the waiting time before the route can be entered), causing alternative routes to be chosen more often. This may even by stimulated by traffic operators using a speed limit measure that increases travel time on overly popular routes. Admittedly, all this is somewhat speculative, but it illustrates that QHM, with its guaranteed travel times, offers a new perspective on the traffic spreading and rerouting problem.

6.5.7 Applications of QHM

Still a lot of research is needed to further corroborate the usefulness of QHM, but until now, we can mention two successful applications of the framework. One deals with a standard for traffic control system interconnection and the other one deals with Coordinated Ramp Metering. Both are described below.

DVM-Exchange

DVM-Exchange (DVM is a Dutch abbreviation for Dynamic Traffic Management) is a standard for the interconnection of traffic control systems, which was developed in The Netherlands in 2012 and 2013 Vrancken et al. (2012). Network Management requires far more connections between control systems than in current, mostly locally operating systems. In order to make these connections as uniform as possible, an interconnection standard is needed, which has a semantic component, expressing the effects on traffic of an exchange between control systems. The description of the semantic component has been based on QHM.

Coordinated Ramp Metering with Priorities

The most common approach to Coordinated Ramp Metering (CRM), the HERO algorithm, has the drawback that it cannot assign priorities to the different on-ramps and to the freeway Papamichail et al. (2010a). When CRM is addressed with QHM, it turns out to be fairly easy to introduce priorities, as was shown by a recent master thesis research Zhi (2014).

6.6 Comparing QHM and the SCM system

In this section, the historical development of the multilevel road network control system is explained first. In 2006, multilevel road network control has been applied for the first time in an experimental implementation: the HARS system (HARS: Het Alkmaar Regelsysteem, the Alkmaar control system in Dutch). Alkmaar is a Dutch city which lies at the end of highway A9 and has a ring road that is often heavily congested. In 2008, the on-ramps and off-ramps of the ring road were re-designed to improve traffic flow. In 2008, the HARS system was upgraded due to these network changes. Since May 2008, a European research project called Control for Coordination of Distributed Systems (C4C, van Schuppen & Villa (2014)) was started to further improve the HARS system. However, the HARS system was not operational anymore since 2010 due to maintenance costs that were too high for municipal and provincial budgets after the worldwide financial crisis. In 2008, the FileProof project was started in order to improve traffic flow and reduce traffic congestion within the Amsterdam area. So both the HARS system architecture and the research results of the QHM framework were applied in the FileProof project to formulate the Scenario Coordination Module (SCM) system. The operational system, which was deployed in September 2010, was evaluated in 2011. The HARS system and the SCM system will be described in detail in chapter 8.

In this section, the SCM system is summarized. In the SCM system, the A10 beltway and urban roads of Amsterdam are divided into 4 parts: the east part, the south part, the west part and the north part. Each part has two directions. Thus the road network is divided into 8 subnetworks. Each subnetwork is predefined by a number of buildingblocks as a part of the configuration of the SCM. If the combination of buildingblock states is changed for a subnetwork after a minimal life time of a scenario (for example 15 minutes), we can conclude that the traffic pattern of the subnetwork has changed. Thus, a new scenario for a subnetwork will be proposed after a minimal life time of the current scenario. The SCM system is also using multi-agent (bottom-up) control. The essence of a multi-agent control mechanism consists of an, often large, number of communicating instances or objects, usually called agents. Each agent has a relatively simple behavior. Usually they all run the same control program or one from a small number of different control programs.

In general, the SCM system is based on the QHM framework with some simplifications and adaptations. These are due to the existing software architecture and constraints of reality. Parts of the SCM system are in line with the QHM framework:

- Both work with expected demand and scenarios. Based on historical data, the expected demand is estimated and expressed in a set of scenarios, including scenarios for special events such as football games.

- Both use a recursive partitioning method to subdivide a given network. To make a given network manageable, the network is geographically split up into parts at several levels.

- Both can handle all kinds of local control measures at each level. Without a serious traffic problem, traffic management remains local at each level.

- Both have coordination between higher and lower levels. In the case of serious traffic problems that can't be handled locally at each level, a higher level is involved to coordinate the lower levels.

- Both combine a top-down control and a bottom-up control. In fact, the bottom-up part consists of real-time, localized adaptations of the top-down control.
- Both can handle different types of road networks, including urban and motorway networks.

On the other hand, the following simplifications and adaptations are made in the SCM system:

- SCM only splits up a network into 4 different levels. QHM can split up a network into an unlimited number of levels.
- Different control levels of the SCM have different control algorithms, whereas QHM has a uniform treatment for different control levels, which means QHM has a generic control algorithm, that applies to all levels.
- SCM uses different control scenarios for a top-down control, however QHM uses different network configuration for a top-down control.
- The basic assumption of omnipresent sensors and actuators, made in QHM, is not true of course. An operational system has to make compromises. SCM focusses on the actually existing sensors and actuators.
- Priority matrices in SCM are only used to resolve conflicts of control scenarios in two adjacent subnetworks. However, priority matrices in QHM are used to distribute the amount traffic flow over the different boundary points.

6.7 Conclusions

The main challenge of NM is making the complexity of road traffic manageable. For this a framework is needed that offers a scalable approach to network management as well as an approach to manage the many connections that NM systems have to make with existing systems in a network. The multilevel traffic control framework combines notions from Systems Engineering with the theory of Hierarchical Control. It is a scalable framework, applicable to networks of any size. It abstracts the behavior of an arbitrary network to effects on the network's boundary. This means that it is applicable to heterogeneous networks (different kinds of roads: urban, motorway, etc.) and that traffic management for networks can be done by managing traffic in a finite set of points.

This approach has been partially tested in both a simulation environment and a practical case study in Amsterdam in chapter 8. With this approach, each control instance, in charge of a part of the total network, has a limited amount of complexity to handle, because it has only a small number of child networks, and it only needs to manage the boundary behavior of its direct children. The internal behavior of each child network is left to the children themselves.

Key algorithms in the multilevel traffic control framework, such as the network partitioning, and determining the coherent network priority configuration for a whole tree of networks, still need further research. The same holds for the exact relationship between entry-exit priorities and child priorities.

Further validation of the QHM framework will take years, which will bring improvements and, no doubt, alternative frameworks.That is no problem. The main message is that a number of key notions, such as scalability, recursion, complexity shielding, hierarchical control, boundary agreements, etc., that are not so common currently in Traffic Engineering, are crucial in order to obtain satisfactory implementations of Network Management for large networks.

Chapter 7

Operational traffic monitoring systems[1]

7.1 Introduction

In a previous chapter, the boundary condition has been explained. In order to perform traffic prediction, the initial traffic states including traffic parameters such as turn fractions need also to be known. As we described in the schematic representation of an ideal Road Traffic Control (RTC) system (figure 1.1) in chapter 1, an accurate, timely and reliable traffic monitoring is the basis of performing on-line traffic prediction. Traffic data collection from road side equipments is essential for traffic monitoring and traffic control. However, traffic data is not always available and, if available, it may has errors. Furthermore, traffic parameters (like turn fraction) which can not be measured directly are also very important for RTC. RTC can be improved if traffic parameters can be estimated as accurate as possible. Last but not least, not a sufficient number of equipments are installed along road side in many places, due to the high hardware cost and the high maintenance cost which make government cannot afford them. In order to solve the above problems in practice, in this chapter, two traffic monitoring applications will be briefly presented by combining the existing approaches with the Bluetooth technology. The traffic monitoring applications are a state estimation application and a Bluetooth tracking application which have been implemented in a distributed architecture presented in chapter 3. These applications have been implemented and tested in an operational system. The author of this thesis was in this chapter responsible for the system design, the part of the system implementation and the major part of the data analyse.

This chapter is organized as follows. Section 7.2 presents a state estimation application to supply missing data and estimate turn fraction by combination of two filtering techniques. Section 7.3 presents an application to measure average travel time by using Bluetooth detectors, in case there are no sensors on motorways. In the end of each

[1] This chapter is based on my publications Wang et al. (2009d, 2011c)

application, the results are presented. Section 7.4 draws overall conclusions for this chapter.

7.2 State estimation by combination of filtering techniques

As we mentioned above, traffic data on motorways is not always available due to the fact that there are no physical sensors or detector failure on some parts of motorways.

In this section, we will present an approach combining Kalman filter and a data filtering technology called the Treiber-Helbing-filter Treiber & Helbing (2002) to contribute to solve the above problems for motorways. First, the Treiber-Helbing-filter is used to estimate the missing data on motorways. Then, the Kalman filter is used to estimate turn fractions on motorways. By applying this approach to traffic control systems, all the missing data and turn fractions can be estimated at the same time.

7.2.1 Estimate Missing Data Using the Treiber-Helbing filter

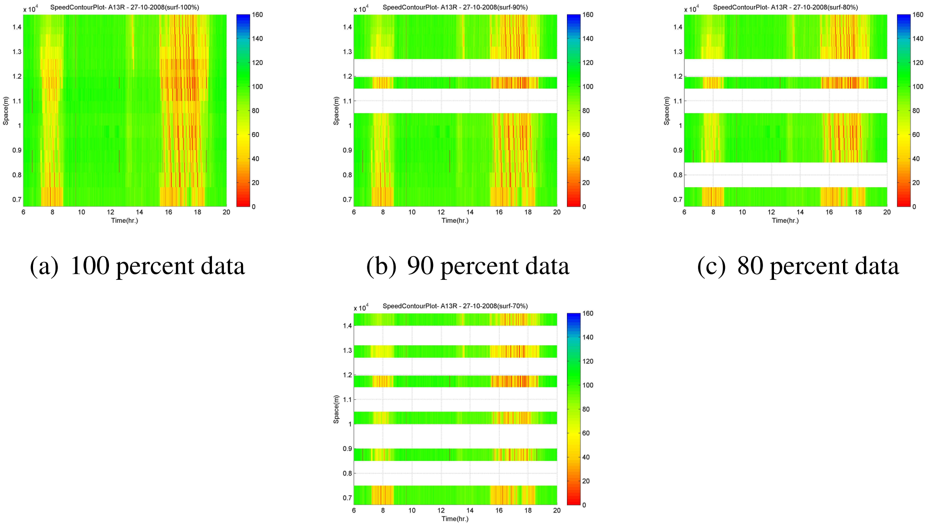
State estimation approaches have been intensively studied in Yuan (2013). In this thesis, the Treiber-Helbing-filter is used to supply data at points with no physical sensors or with detector failure. The Treiber-Helbing-Filter is a spatio-temporal traffic state estimator. The basic idea of the Treiber-Helbing filter is to calculate each point over time and space with a linear combination of the neighbour measurement data Treiber & Helbing (2002).

In order to evaluate the effects of the Treiber-Helbing filter on data completion, we take raw data of a day (from 6:00 to 20:00 in one-minute pattern) from a Dutch motorway (A13) as example. The current monitoring system on the Dutch motorways consists of double inductive loops located about every 500 meters. Some test scenarios with different extent of data missing are presented in table 7.1. 100 percent data means that all the data of the double inductive loops is available; 90 percent data means that 90 percent data of the double inductive loops is available, etc.

7.2.2 Results: Estimate Missing Data

To make a comparison between different scenarios, the filtering result derived from the overall temporal-spatial filtering based on the complete data set, is used as reference. The performance measures used are the same as the measures presented in van Lint & Hoogendoorn (2009).

The speed contour plots with both raw and filtered information are shown in figure 7.1 and figure 7.2 respectively. It is useful to point out the difference between the speed



(a) 100 percent data

(b) 90 percent data

(c) 80 percent data

(d) 70 percent data

Figure 7.1: Data missing

Table 7.1: Comparison results for data missing.

Data missing level		90 percent	80 percent	70 percent
Speed	RMSE(km/h)	1.3262	1.7342	1.9501
Flow	RMSE(veh/h)	94.7103	124.7060	151.1307
Speed	MPE(%)	0.3737	0.2864	0.3911
Flow	MPE(%)	0.7337	0.3653	0.5920
Speed	MAPE(%)	0.8044	1.2065	1.5875
Flow	MAPE(%)	0.9301	1.4881	2.1758
Speed	SPE(%)	3.0248	3.6815	4.1185
Flow	SPE(%)	2.3842	3.1907	3.8859

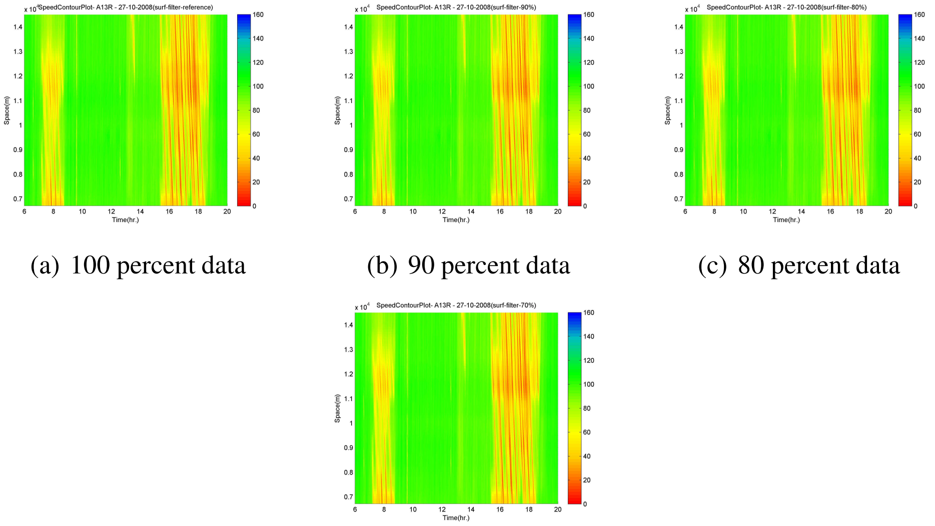
contour plots with data missing and the reference case. In table 7.1, the comparison results are presented. It is found that even if data are missing of 30 percent of the detectors, the filter still yields acceptable results. The meaning of the different error indicators can be read in table 7.2

In case of missing data, the filter turns out to be efficient to rebuild lost data. Moreover, the relative error indicators between the filtered estimated data and the filtered original data are rather small, as shown in table 7.1.

In general, the Treiber-Helbing filter is able to estimate missing data. The estimated data are acceptable in terms of the selected error indicators.

Table 7.2: The meaning of different error indicators.

Error indicator	Meaning
RMSE	The root mean of squarred error
MPE	The mean percentage error
MAPE	The mean absolute percentage error
SPE	The standard deviation of the percentage error



(a) 100 percent data

(b) 90 percent data

(c) 80 percent data

(d) 70 percent data

Figure 7.2: Data filtering

7.2.3 Turn Fraction Estimation Using the Kalman Filter

The literature offers an abundance of approaches to estimate turn fractions, such as Likelihood methods Nihan & Davis (1987), Bayesian estimator Zijpp (1997) and Kalman filter Cremer & Keller (1987). The most common approach is based on the Kalman filter. We mention only a few approaches. Estimation of turn fractions based on the unconstrained and constrained Moving Horizon Estimation (MHE) Kalman filtering is presented in Kulcsar & Varga (2004). Traffic state estimation based on the extended Kalman filter is presented in Wang et al. (2008). Most studies are simulation based.

In this thesis, the Kalman filter is used to estimate turn fractions of motorways in the Netherlands such as the A7-A8 crossing which can be seen in figures 7.4. The red links are the mainlinks which are connected with two junctions. The blue links are the accessorlinks which are connected with mainlinks.

Model Contruction

The above junction can be simplified as the junction in figure 7.3. If we consider the flow of the incoming mainlinks as q_i, the flow of the outgoing mainlinks as y_j and the turn fractions of accessorlinks as $x_{ij}(0 \leq x \leq 1)$, the following euqations can be derived:

$$y_1 = q_2x_{21} + q_3x_{31} + q_4x_{41} \tag{7.1}$$

$$y_2 = q_1x_{12} + q_3x_{32} + q_4x_{42} \tag{7.2}$$

$$y_3 = q_1x_{23} + q_2x_{23} + q_4x_{43} \tag{7.3}$$

$$y_4 = q_1x_{14} + q_2x_{24} + q_3x_{34} \tag{7.4}$$

$$1 = x_{12} + x_{13} + x_{14} \tag{7.5}$$

$$1 = x_{21} + x_{23} + x_{24} \tag{7.6}$$

$$1 = x_{31} + x_{32} + x_{34} \tag{7.7}$$

$$1 = x_{41} + x_{42} + x_{43} \tag{7.8}$$

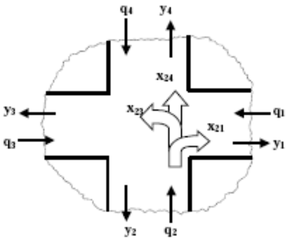


Figure 7.3: Schematic representation of a simplified junction.

Based on the above equations, the junction can be modelled as the following state space modelKulcsar & Varga (2004); Kulcsar et al. (2005):

$$x_{k+1} = Ax_k + \omega_k \tag{7.9}$$

$$y_k = C_kx_k + \nu_k \tag{7.10}$$

Where $x_k = \begin{pmatrix} x_{12} & x_{13} & x_{14} & x_{21} & x_{23} & x_{24} & x_{31} & x_{32} & x_{34} & x_{41} & x_{42} & x_{43} \end{pmatrix}$, $y(k) = \begin{pmatrix} y_1 & y_2 & y_3 & y_4 & 1 & 1 & 1 & 1 \end{pmatrix}, A = I$

and $C_k = \begin{pmatrix} 0 & 0 & 0 & q_2 & 0 & 0 & q_3 & 0 & 0 & q_4 & 0 & 0 \\ q_1 & 0 & 0 & 0 & 0 & 0 & 0 & q_3 & 0 & 0 & q_4 & 0 \\ 0 & q_1 & 0 & 0 & q_2 & 0 & 0 & 0 & 0 & 0 & 0 & q_4 \\ 0 & 0 & q_1 & 0 & 0 & q_2 & 0 & 0 & q_3 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$ with x_0 given. ω_k is the state noise vector, ν_k is the measurement noise vector.

If the flow of the incoming and outgoing mainlinks are known, the turn fractions of accessorlinks can be estimated by applying the Kalman filter.

From the above description, we know that the traffic information(flow) from all mainlinks are needed for the turn fraction estimation. However, the traffic information from the mainlinks of motorways is not always available due to the fact that either there are no physical sensors or there are detectors failure. Thus, the Kalman filter cannot be applied. The Treiber-Helbing-filter which has been explained above is able to solve the above problem for motorways.

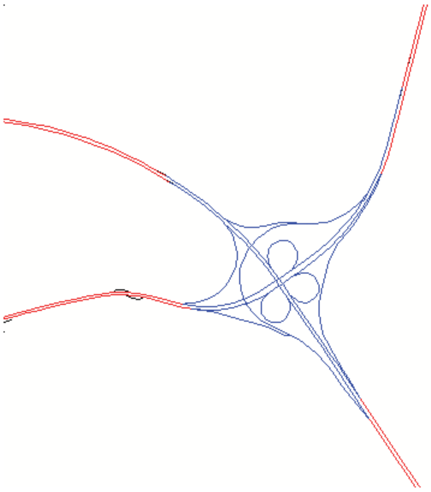
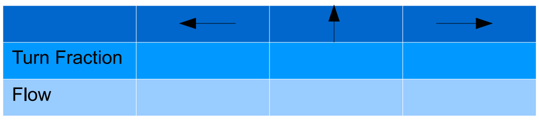


Figure 7.4: A7-A8 crossing

7.2.4 Results: Turn Fraction Visualization

Based on the data estimated by the Treiber-Helbing-filter, the turn fractions can be estimated by using the Kalman filter and be visualized in traffic control software.

An automated image of the junction is created based on the image of links. This image and a table with turn fraction are shown in a graphical user interface. Clicking in the junction image will result in an overview from this image containing the information presented in the tables (Fig. 7.5).



	←	↑	→
Turn Fraction			
Flow			

Figure 7.5: Table of turn fraction

7.2.5 Conclusion

In this section, we described our contribution to estimate missing data and turn fractions at the same time by combining the Kalman filter and the Treiber-Helbing-filter. It is very important to estimate the traffic parameters for the purpose of traffic control. The Kalman filter is a common approach to estimate turn fractions, however the Kalman filter cannot always be applied due to the fact that the sensor information from mainlinks of motorways is not always available. The Treiber-Helbing-filter can contribute to solve the problem by estimating missing data of mainlinks. Then, the Kalman filter is able to estimate turn fractions with traffic information of all mainlinks.

7.3 Measuring travel time by Bluetooth tracking

Traffic data collection from road side equipments is essential for traffic management, the more accurate traffic information, the better the performance of traffic control. The common traffic data detectors are loops, cameras with license plate recognition and radars, however they have either high hardware costs or high maintenance costs. GSM and GPS signals are also good sources to collect traffic data, however the data is not sufficient to estimate the traffic state. Thus, recently a new type of detector was taken into use, bluetooth detectors can lower the high hardware costs and the high maintenance costs. Bluetooth detectors are attractive alternative equipments to collect traffic data with relatively low cost.

Elsewhere Bluetooth data collection is also becoming very popular in traffic research and in operational systems, both for in-vehicle driver support and for roadside traffic data collection Barcelo et al. (2010), Haghani et al. (2010), Malinovskiy et al. (2010). The data collected are mostly travel times and travel time variance, but also dynamic OD matrices are estimated on the basis of Bluetooth data.

There are many advantages in using bluetooth detectors to measure traffic data.

- High detection rate: Nowadays, almost everyone has a mobile phone with a Bluetooth device. Each Bluetooth device has a worldwide unique address; this address is detectable. Approximately 20 to 30 percentage of road users can be detected by using the bluetooth detector, so that we can receive guaranteed, accurate travel information. Comparing with the Bluetooth detectors, camera's are not so reliable if weather conditions are bad. Moreover, more and more Bluetooth devices will be used, so the detection rate is expected to increase.

- Very cheap: A Bluetooth receiver only costs about 10 Euro. We can plug it in a mini-computer, then it can be used to collect traffic data. Complete unit cost is much cheaper than the loop detectors and camera's if deployed on a large scale.

- Relative low maintenance costs: The device is very simple to install and it is also very easy to replace.
- Sufficient travel information of traffic state: Travel time of individual road users are measured, thus the received traffic information is not fragmented and it can represent the real traffic state.

There is a privacy problem with Bluetooth, but it can be handled by storing Bluetooth IDs in an anonymous way. Because the relation of a person or vehicle with the bluetooth address is not known, the anonymity is guarded.

With so many advantages, Bluetooth detectors are used to measure travel time on freeway in stead of camera's or induction loops. In this chapter, we will present travel time measurements by using Bluetooth detectors and an algorithm to calculate the average travel time based on the measurements. The work has been done by the auther of this thesis in 2010. After that, there are more publications about Bluetooth detectors.

7.3.1 Test in Belgium

In order to evaluate the accuracy of Bluetooth detectors for travel time measurements on mortorways, a test in Belgium was done from 26-08-2010 till 07-10-2010 on the freeway E313 (A13) between Geel and Antwerp. Four Bluetooth detectors are placed near ramps (See Figure 7.6). Figure 7.7 shows how the Bluetooth detector looks like in this test.

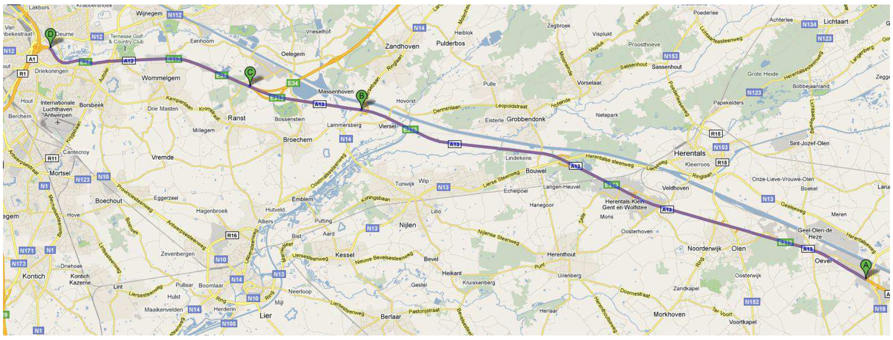


Figure 7.6: Four Bluetooth detectors.

The total distance between point A and point D is around 36 km with a free flow travel time of 23 minutes. The total distance divided by the maximum velocity is the free flow travel time. Table 7.3 shows the distance and free flow travel time between four points.

The work principle of the Bluetooth detectors has been explained in many papers Barcelo et al. (2010), Haghani et al. (2010), Malinovskiy et al. (2010). In the thesis,

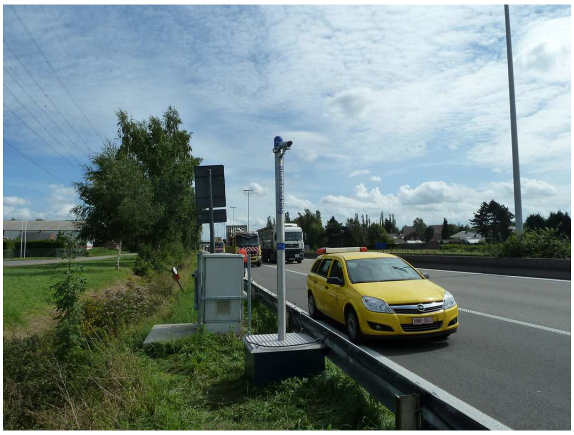


Figure 7.7: Bluetooth detector.

Table 7.3: Distance and free flow travel time.

Detectors	Distance (km)	Free flow travel time (min)
A-B	22	14
B-C	5	3
C-D	9	6

we just give a short summary. We receive a list of Bluetooth data with Bluetooth IDs for each measurement point every minute. Then we try to match the Ids at two measurement points. Once a match is found, travel time for the road user can be calculated. Figure 7.8 and Figure 7.9 show the measured travel time for each road user by using Bluetooth detectors. For the long route, many road users with high travel time are

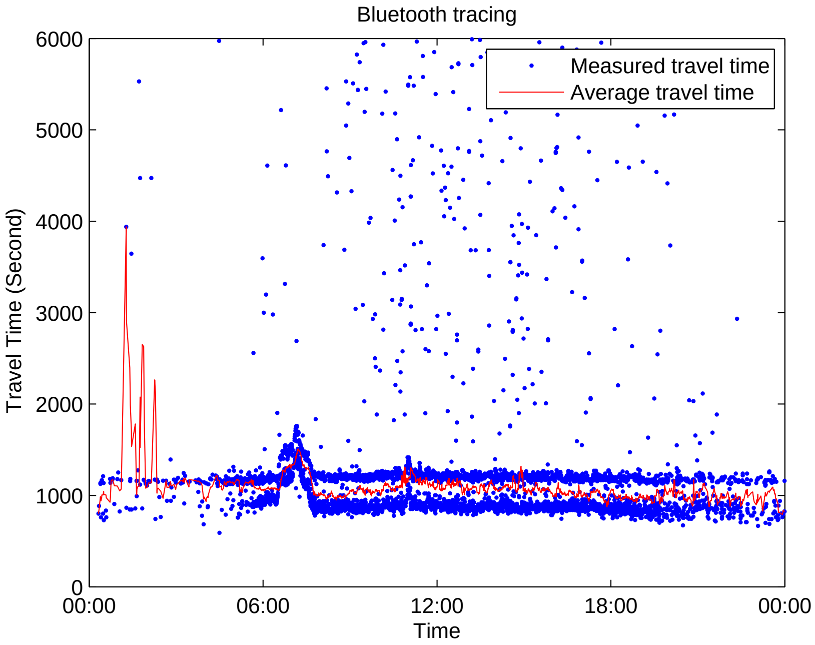


Figure 7.8: Travel time from point A to point B.

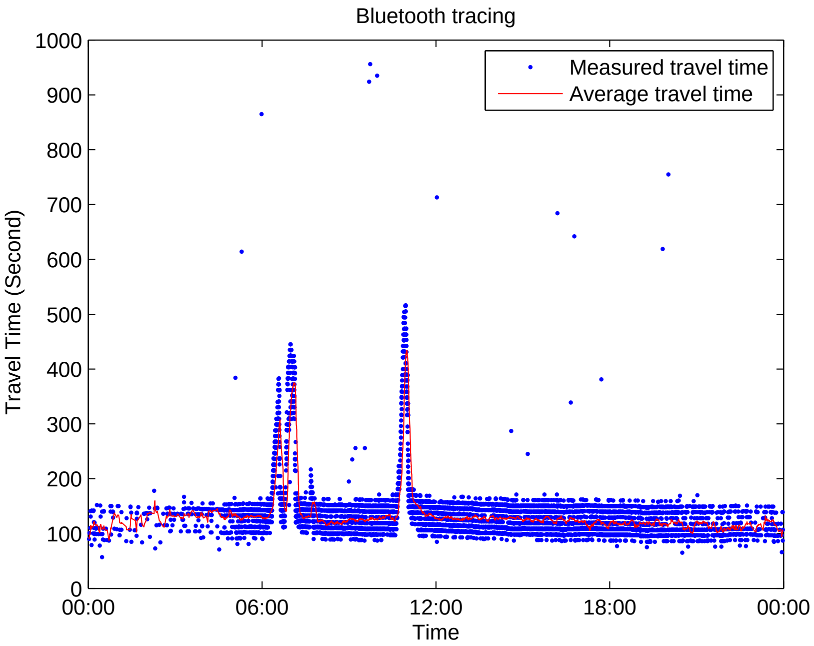


Figure 7.9: Travel time from point B to point C.

measured. This is due to the fact that there are gas stations in the middle of the route. Road users might stop at the gas stations and continue their trips after some while. There are not too many data with high travel time for the short route, since there are no gas stations or entry and exit points in the middle of the short route.

Another interesting fact is that 3 to 10 road users are measured every minute for the long route and 8 to 20 road users are measured every minute for the short route. The reason is that there is no way out for the short route, so the road users detected at the starting point of the route are also detected at the end of the route mostly. However, the road users might exit in the middle of the long route. Thus, the detection rate for the short route is higher than the long routes.

7.3.2 Algorithm: average travel time calculation

As we discussed in the previous section, there are many hits with high travel time which are outliers. Thus, in order to calculate the correct travel time, we need to remove the outliers. In this chapter, a simple algorithm is used to remove outliers and a moving average algorithm is used to calculate the average travel time.

Input detections are stored in an input table until a configured max-time (for example 2 hours) is reached. The configured max-time can bound the measured travel time and remove outliers. Output detections are looked up in the input table at the moment they are received. If a match is found, the difference in time is stored in a match table. If the time is negative, it is not stored in the match table. Every minute, the match table

Table 7.4: Results of Bluetooth data

	Free flow	Congestion	Heavy congestion
MAPE	7.5%	3.5%	2.8%

of the recent 10 minutes is analyzed. First, outliers are further removed by a simple algorithm. All hits that are more than 50 percent above the average are removed form the table. As long as there are values above this threshold, the average is re-calculated and the high values will be removed. Then, the average travel time is calculated by a moving average algorithm. The average travel time for each minute is the mean value of the recent 10 minutes measured travel time.

7.3.3 Results and analysis

The average travel time is shown on the above figures. From the figures, we can see that there are the two lines on the long routes. It might be due to the fact that there are two lanes on the routes, cars on the left lane are driving faster than cars on the right lane. For the long route, the travel time during congestion at 07:00 AM is 1.5 times the travel time of free flow. For the short route, the travel time during congestion at 7:00 AM and 11:30 AM is 4 times the travel time of free flow.

The estimated average travel time in the beginning and the end of the day shows significant peaks. The peaks are not consistent with the travel time measured by loop detectors and camera's, thus we can conclude that the average travel time is calculated incorrectly during the period. The reason is that there are too little Bluetooth measurements, thus the outliers cannot be removed properly. However, there is no peak for the average travel time of the short route in the beginning and the end of the day. It is due to the fact that there are relative more hits per minute for the short route, thus the outliers can be removed easily.

While travel time is measured by Bluetooth detectors, traffic information for the same period is also collected by Automated License Plate Recognition (ALPR). If we consider the ALPR data as the ground truth, the MAPE of Bluetooth data is shown in table 7.3.3.

The reasons why the bluetooth data are different from the ALPR data are:

1. The location of bluetooth detectors are not exactly the same as the camera detectors.

2. The sample sizes are 250 times differences: 25 frames per second for camera (means the sample size of camera is 1/25 second), however the sample size of bluetooth is 10 seconds.

3. The significant differences between the travel time distributions collected by both devices: in other words, we measured much more travel time with very high values by using Bluetooth than camera's. The outliers cannot easily be removed during the average travel time calculation.

4. The detection area of Bluetooth device(200 meters) is much larger than camera's(20 meters): the exact location of vehicles which detected by Bluetooth device is much uncertain than the exact location of vehicles which detected by camera's.

5. In case of the traffic jam, the travel time will be longer. Then the impact of the above difference will be less and the difference between Bluetooth and camera will be also less.

Actually, the ALPR data is not the ground truth (means it can not represent the real travel time), since camera's also have some bias in travel time measurement. So the difference between ALPR and bluetooth doesn't mean the ALRP data is better than Bluethooth data. The GPS data is also compared with the Bluetooth data in Haghani et al. (2010). The results show that the travel time measured by Bluetooth detectors is not significantly different from the actual travel times measured by the GPS sensors.

Compared with the Bluetooth test on urban roads van Zuylen et al. (2010), travel time measured by Bluetooth detectors is much reliable for both short routes and long routes on a freeway. The reason is that the distance between on-ramp and off-ramp is relatively large. Thus, it is relatively easy to remove the outliers, since, compared with urban traffic, the chance that a car makes a U-turn is very small.

7.3.4 Conclusions

We present travel time measurements by using Bluetooth detectors on freeways and the algorithm to calculate the average travel time. The average travel time measured by Bluetooth detectors, has been compared with traffic time which measured by ALPR. For both the short route (5 km) and the long route (>10km), results are quite good compared with the Bluetooth test on urban road. Furthermore, the reliability of Bluetooth detectors are higher than loop detectors and camera detectors. Thus, Bluetooth detectors are cheap and reliable alternative sensors on freeways.

7.4 Conclusions

In this chapter, two traffic monitoring applications are presented. All the applications are based on the same distributed architecture which is described in Chapter 3. In case of sensors such as loops, cameras with license plate recognition, radars, GSM and GPS are available to collect traffic data, the Treiber-Helbing-filter and the Kalman filter are combined to estimate missing data and turn fractions at the same time. The results show that missing data is reconstructed efficiently with relatively small error. In case there are no sensors on motorways, Bluetooth detectors can be used to measure the average travel time. Furthermore, the detection rate of Bluetooth devices is increasing. The results show that Bluetooth detectors are a cheap and reliable alternative sensors on freeways.

Chapter 8

Operational traffic control systems[1]

8.1 Introduction

In a previous chapter, distributed traffic prediction has been explained based on the distributed architecture of chapter 3. The traffic prediction can be used as a decision support tool to apply or to evaluate traffic control. In chapter 6, the framework of a multilevel control has been explained. In this chapter, the distributed architecture and the multilevel control framework (with some adaptations to accommodate the real-time applications) will be used for traffic control in two practical case studies. The best known approach Rathi (1988) in traffic network control is the one characterized by traffic management centers applying so-called scenarios in response to current and predicted traffic states of the road network. Scenarios are a coherent set of local measures for a road network. They are developed off-line and correspond to recurring patterns in the traffic state, such as the morning rush hour or the weekend exodus. We will call this the top-down approach, as the traffic management center is the only entity allowed to take decisions. The top-down approach has been used in the Netherlands for the last decade; it is a generic, but limited, way of controlling the overall traffic performance in a network.

The scenario approach is confronted with a number of problems, which in recent years became apparent by the development of scenarios for the Amsterdam area. First of all, due to the size of the network considered and the many states in which this network can be, the number of scenarios needed may easily grow beyond a practically manageable set. In addition, there are several different traffic management authorities active in the area, each of which starts by developing its own set of scenarios for its own part of the network (motorways, urban, provincial). Yet, networks are connected, so different scenarios from different authorities can easily create conflicts on the boundaries between different networks. As a third problem, even a large set of scenarios still doesn't come close to the many states in which traffic can be, since the scenarios are optimized to

[1] This chapter is based on my publications Wang et al. (2009a, 2010c,b, 2011b, 2012c)

a specific traffic pattern, which may differ from real cases. There was still a need for local, real-time adaptations of scenarios to local traffic states.

In order to solve the above problems, the focus in this chapter is to explain two distributed control systems which have been implemented in two different projects. The distributed control systems are a system with integrating Top-down and Bottom-up Control in the HARS project and a system with distributed scenarios over network in the FileProof project. These systems have been tested in an operational system. The author of this thesis was in this chapter responsible for the system design, the core part of the system implementation and the major part of the data analyse.

This chapter is organized as follows. Section 8.2 presents the traffic control system for the HARS project. Section 8.3 presents the traffic control system for the FileProof project. Afterwards, the performance of the control system is presented. Section 8.4 draws overall conclusions for this chapter.

8.2 Integrating Top-down and Bottom-up Control in the HARS project

Alkmaar is a city in the Netherlands which lies at the end of highway A9 and has a ring road that is often heavily congested. The network management hierarchy will serve in this project as a complement to regular dynamic traffic management solutions (especially control schemes).

In this project there are three types of sensors: speed-flow measurement points, the induction loops at TLS (Traffic Light System) sites, and a traffic simulation model called MaDAM Mad (2006). The induction loops and measurement points are sensors that collect real-time traffic information from the environment, such as the velocity. MaDAM acquires the information from the TLS loops and velocity-intensity measure points and determines what the traffic state is on links that have no sensors of their own. In addition, MaDAM predicts what the traffic states will be for the links in the next 30 minutes in blocks of 5 minutes. As a matter of fact, the use of real-time data combined with historical statistics enables the system to be predictive, which enhances the efficiency of the traffic.

In this project there are two types of actuators: the TLS and the DRIP (Dynamic Route Information Panels). The TLS actuator reacts based on information sent by the links. They are used to reduce or increase the intensities of traffic flows. The DRIPs are used for rerouting and informing drivers on the current traffic state of routes downstream of the DRIP. In case of congestion, drivers will observe the extra travel time and from this they can estimate the volume of congestion. Knowing what to expect adds to the driver's comfort level. More importantly, route information enables drivers to make a different route choice. Drivers are thus diverted away from the congested road with positive effects on its congestion. The extent to which this rerouting occurs is small though. In fact, it is often estimated that the amount of drivers that alter their route choice based on information provided by a DRIP is at most 15% Alk (2003). Nevertheless, DRIPs are a valued and widely deployed instrument. In the HARS project, the functions of the Origin-Destination managers (ODMGRs) are enhanced and extended to enable more effective rerouting. Much traffic needs to pass over the ring. The use of the DRIPS enables RTC to influence the side which is used (e.g. East or West). If one side of the ring is (nearly) congested, traffic can thus be diverted to the other side.

8.2.1 Top-down control: Control schemes

In this project, top-down control is performed by control schemes. Five different control schemes for morning rush hour (MRH), afternoon rush hour (ARH), Friday (FRI), Sunday (SUN) and quiet hour (QH) are used as examples to show how the top-down control works. Morning rush hour is from 6:30 to 9:30. Afternoon rush hour is from 4:00 to 7:00. Quiet hour is the time excluding morning rush hour, afternoon rush hour, Friday and Sunday.

In this project, the control schemes are based on eleven Origin-Destinations (OD pairs) which are shown in table 8.1. We can easily know the origin and the destination from the name of ODs, for example the origin and the destination of "A9_N9_DEN_HELDER" are the Dutch highway A9 and provincial road N9 to the town of Den helder, respectively. The different ODs of the Alkmaar belt can also be seen from figure 8.1. The numbers in the table stand for the expected traffic flow in number of cars per hour. These numbers are determined based on the real traffic situation in the five different cases. And the corresponding set of optimal control signals for TLS are also designed. In the on-line case, the best matched control scheme compared with the current traffic states is selected to control TLS automatically.

$$ODMatch = 100(1 - \frac{1}{n}\sum_{i=1}^{n}|\frac{ODe_i - ODc_i}{ODe_i}|) \tag{8.1}$$

For example, the OD values for the traffic states in Monday morning 8:00 are 9750, 246, 190, 717, 1848, 1803, 201, 314, 1971, 229 and 695, respectively. We calculated the match of OD values based on equation (8.1). In the equation, ODe is the expected OD value, ODc is the current OD value and n is the number of ODs. The results of the OD match are shown in table 8.2. In the case all values of OD match are relatively low (for example lower than 70), it is better to switch on control scheme based on the time interval. Thus the time match is introduced. If the current time is in the time interval of one control scheme, the time match of the control scheme is equal to 70 and the others are equal to 0.

The final decision is made based on the OD match and the time match. If the biggest

Table 8.1: Average flow per period for sereval OD pairs.

OD	MRH	ARH	FRI	SUN	QH
A9_N9_DEN_HELDER	11080	23110	21980	25130	11740
TERBORCHLAAN_N9_DEN_HELDER	200	700	540	170	230
BERGERWEG_TERBORCHLAAN	170	360	330	110	150
N9_DEN_HELDER_TERBORCHLAAN	640	280	220	70	210
BERGERWEG_KENNEMERSTRAATWEG	1650	2670	2750	1790	1450
KENNEMERSTRAATWEG_BERGERWEG	1610	1840	1980	1630	1130
TERBORCHLAAN_BERGERWEG	180	260	310	130	130
TERBORCHLAAN_KENNEMERSTRAATWEG	280	380	440	230	260
N9_DEN_HELDER_KENNEMERSTRAATWEG	1760	950	2930	650	930
KENNEMERSTRAATWEG_TERBORCHLAAN	260	380	420	200	220
KENNEMERSTRAATWEG_N9_DEN_HELDER	790	148	1300	1170	800

Table 8.2: Match values

	MRH	ARH	FRI	SUN	QH
ODMatch	88	17	45	0	47
TIMEMatch	70	0	0	0	0

value of the OD match is higher than the biggest value of the time match, the corresponding control scheme of the OD match will be switched on. Otherwise the corresponding control scheme of time match will be switched on. As a matter of fact, the control scheme for the morning rush hour will be switched on based on the calculation in the table 8.2. The principle of the top-down control can also be shown in figure 8.2.

The control schemes are optimized to a specific traffic pattern, which may differ from real cases. The number of control schemes are limited, which cannot optimally deal with all kinds of traffic situation. Due to these draw-backs of top-down control approach, the bottom-up control approach is integrated to improve the performance of traffic control system in this project.

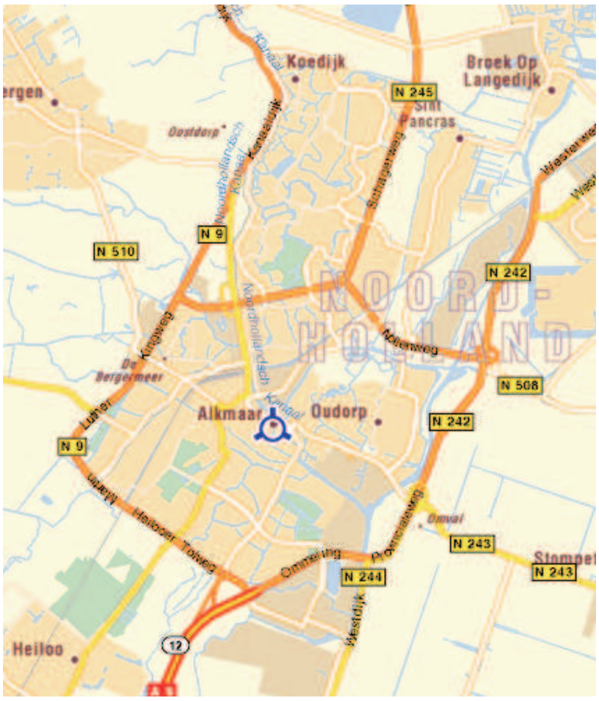


Figure 8.1: The Alkmaar belt.

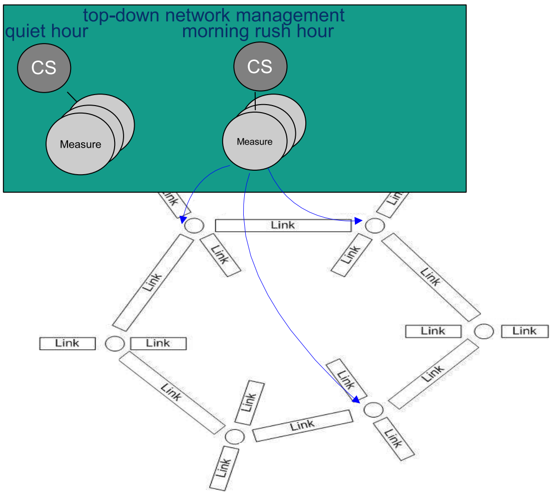


Figure 8.2: Top-down control.

8.2.2 Bottom-up control: Service calls

In this project, bottom-up control is performed by so-called *capacity service* which actually are actions configured in link-agents. By using capacity services call, the links are able to offer and ask for capacity from downstream, upstream links respectively.

More precisely, links compare the information about their traffic state with a so-called "*reference framework*" Rij (2003). The reference framework describes the criteria that the traffic state on the links should ideally meet. If the traffic state differs from the criteria links will take appropriate action. The links (mainlinks or accessorlinks) will communicate with other links and ask them to perform appropriate services to induce a traffic state that meets the criteria. The first capacity service to be implemented is called "reduce outflow", which is offered by all links. When called upon, a link will reduce its outflow by setting different green times for the TLS. If the link does not have a TLS (or any other way to implement the service) it will forward the service-call to his upstream neighbouring link(s). Also, when the link calculates that the reduction in green time (and thus in outflow) will make it miss its own reference framework, the link will send a corresponding "reduce outflow" request to its predecessors. This reactivity is one characteristic of an agent and makes the system intelligent. If there is more than one upstream link, the link will divide the request among the links. In this division, the traffic states on the upstream links as well as their priorities within the network are taken into account. First, links with remaining buffer-capacity will be allocated as much of the service request as their buffer capacity allows. Second, links with higher priority only have to realize a smaller portion of the outflow reduction.

The principle of bottom-up control can also be shown in figure 8.3. In the case an accident happened in a link, the capacity services in the link are called to reduce outflow from his upstream links. By doing so, the bottom-up control approach can dramatically improve the performance of the traffic control system, since the top-down control approach can only guide the road user rerouting by using DRIPs which have less contribution to solving the traffic problem.

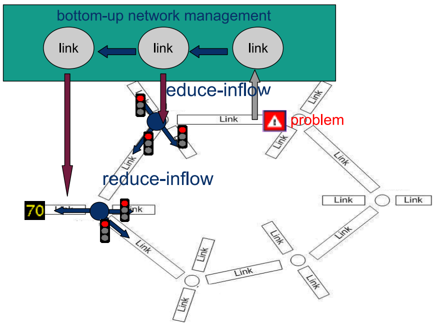


Figure 8.3: Bottom-up control.

8.2.3 Conclusions

In this thesis, we presented an approach combining both top-down and bottom-up control in one control strategy using hierarchic agents for road network control. The approach is currently being implemented in one case study: the HARS system, which was developed at Trinité Automation B.V. The case study has demonstrated that a bottom-up approach to traffic control is a promising addition to the top-down approach. In fact, the top-down and the bottom-up control will not replace each other. The combined approach can have all the advantages of both approaches and avoid their disadvantages. It is able to perform traffic control from both traffic control central (the top) and link-agents (the bottom). It leads traffic situation to both a global and local optimum. Thus, the combined approach improves performance and efficiency of the traffic control system.

8.3 Traffic Control in the FileProof project

Before the FileProof project started, urban and freeway traffic networks were controlled separately by the municipality of Amsterdam and by the highway agency. Both of them try to get rid of congestions and optimize traffic flows from their own point of view. As a result, there are lots of conflicts between the controllers of the urban network and freeway network. Moreover, all roadside equipment is controlled independently, for example, different systems are used to control traffic lights, ramp metering, etc. So traffic operators have to switch between different systems in order to control traffic. It is huge amount of work for operators and it also gives more chances for operators to make mistakes in their daily work.

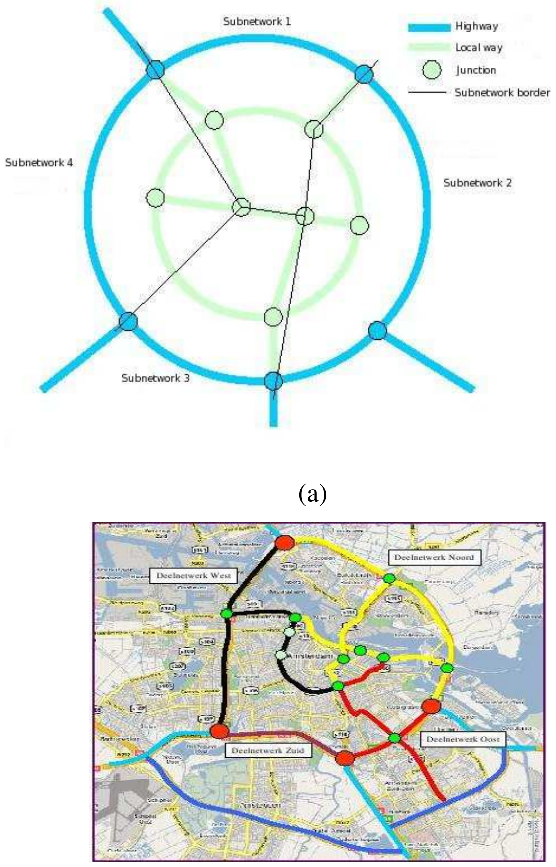
In order to improve traffic flow and reduce traffic congestion within the Amsterdam area, the FileProof project started in 2008. In this project, traffic engineers defined a network control strategy for the A10 beltway and urban roads of Amsterdam, which results in more than 700 scenarios for the scenario-based traffic control. However, it is almost impossible for operators to handle so many scenarios and select the best one suitable for the current traffic pattern. The SCM has been introduced to address this problem with a bottom-up approach. It is the first time to integrate urban traffic control and freeway traffic control in one system in Holland.

In this project there are four types of sensors: MONICA sensors (velocity-flow measure points with double induction loops), the induction loops at ramp metering sites, MOCO sensors (travel time measure points by using camera's with license plate recognition), and the induction loops at TLS (Traffic Light System) sites. The induction loops and measure points are sensors that collect real-time traffic information from the environment, such as velocity and flow.

In this project there are four types of actuators: the TLS, the ramp metering system, the text DRIP (Dynamic Route Information Panels) and the image DRIP. The TLS actuators are used to reduce or increase the intensities of traffic flows on urban roads. The ramp metering uses traffic signals at freeway on-ramps to control the traffic flows entering the freeway. Both the text and image DRIPs are used for rerouting and informing drivers on the current traffic state of routes downstream of the DRIP. In case of congestion, drivers will thus know the extent of congestion that they can expect. Knowing what to expect adds to the driver's comfort level. More importantly, route information enables drivers to make a different route choice. Drivers are thus diverted away from the congested road, with positive effects on its congestion.

8.3.1 Definition of subnetworks and buildingblocks

In this project, the A10 beltway and urban roads of Amsterdam are divided into 4 parts: the east part, the south part, the west part and the north part. Each part has two directions. Thus the road network is divided into 8 subnetworks (figure 8.4). The boundary of each subnetwork can be seen in figure 8.4, the different colors express the different subnetworks.



(a)

(b)

Figure 8.4: Network and subnetworks of Amsterdam area.

Each subnetwork consists of 3 to 5 buildingblocks consisting of either freeway or urban roads. A buildingblock is defined by a number of links to provide traffic data from all kinds of sensors. Each buildingblock has three different states: the green state (free flow), the yellow state (in between) and the red state (congestion). Different traffic services are defined for each traffic state. Both buildingblocks and subnetworks can be pre-defined by a configuration tool

8.3.2 Working principle of the SCM

After subnetworks and buildingblock are defined, the state of buildingblocks can be calculated based on the traffic states of links every minute. And the corresponding traffic measures can be used. The combination of the traffic measures in buildingblocks can represent a scenario of a subnetwork. If the combination of buildingblocks states is changed for a subnetwork after a minimal life time of a scenario (for example 15 minutes), a new scenario will be proposed by the SCM system. Operators can accept or decline the scenario. If the scenario is declined, the scenario of the subnetwork remains the latest activated scenario. Operators can also manually change the scenarios for buildingblocks in subnetworks as they want. Moreover, the performance of the subnetwork can be evaluated by the states of buildingblocks. More buildingblocks with green indicator values in the subnetwork means a better performance of the subnetwork. Each subnetwork is independent and can have its own scenarios. Thus the combination of subnetworks can represent a scenario of a network.

The only problem is that there might be conflicting services at the boundary between adjacent subnetworks. The potential conflicting services are shown in figure 8.5. Buildingblocks can communicate with other buildingblocks to resolve a conflicting service by themselves based on their priority. The priority of buildingblocks in the conflicting location is different for morning and evening rush hours. The left figure shows priorities in the potential conflicting locations for the morning rush hour; the right figure is priorities in the potential conflicting location for the evening rush hour. The buildingblock with higher priority (low value) can use its service to control the traffic. The one with lower priority (high value) has to disable its conflicting service.

The bottom-up control which was explained in section 8.2.2 is still functional in the multi-agent control layer.

8.3.3 Performance evaluation

In order to clearly see the performance of the SCM system, the author of this thesis did a quantitative evaluation recently to measure the effects on the A10. We took the traffic data from two different days for the analysis. One of the day is November 7th, 2011, which is an ordinary Monday, having only local traffic control without network coordination. The other day is November 14th, 2011, which is also Monday, but with the

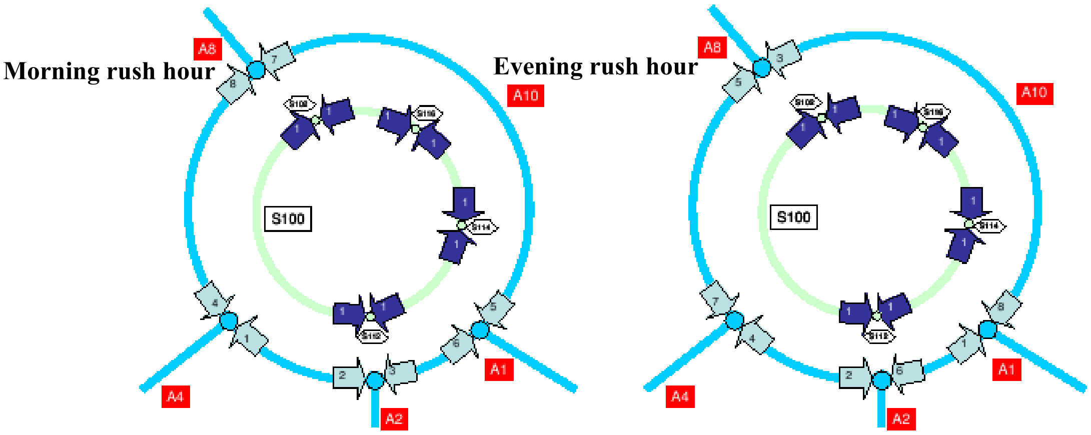


Figure 8.5: Conflicting services for morning rush hour and evening rush hour.

SCM in operation. To illustrate the traffic flow of the whole network, both directions of the A10 ring around Amsterdam city are measured here.

Figure 8.6 illustrates how the travel time around the full inner ring varies within 24 hours. The blue line is the travel time before using the SCM and the red line is the travel time after using the SCM. In this figure, it is also shown that traffic during the morning rush hour is much better than during the evening rush hour. Of course, the positive effect of introducing the SCM is a lot more obvious in the evening rush hour. The travel time of the evening rush hour is zoomed in in Figure 8.7, so that the differences between with and without applying the SCM become more apparent. Figure 8.8 and Figure 8.9 illustrate how the travel time of the outer ring varies within 24 hours and during the evening rush hour respectively. The result in these figures clearly demonstrate that applying the SCM effectively smooths the traffic flow on the ring road around Amsterdam city. The average delay rate (ADR) expresses how

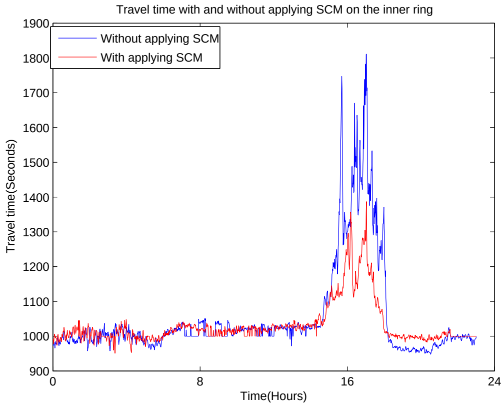


Figure 8.6: Travel time for the inner ring.

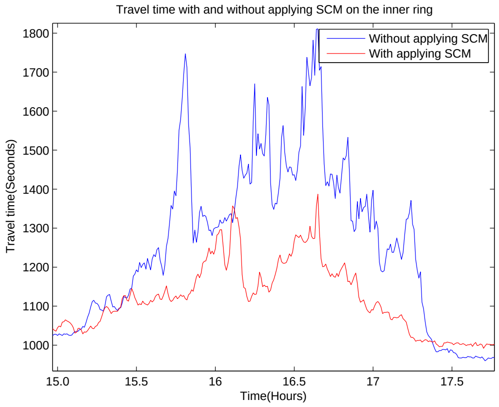


Figure 8.7: Travel time in the evening rush hour for the inner ring.

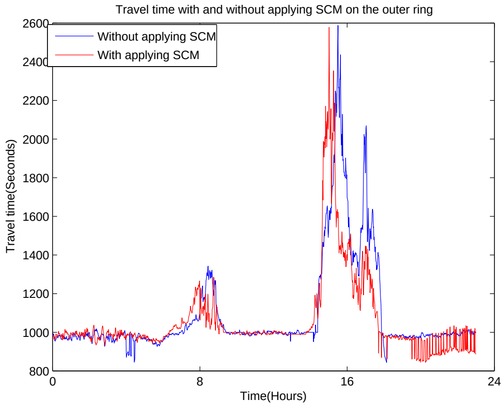


Figure 8.8: Travel time for the outer ring.

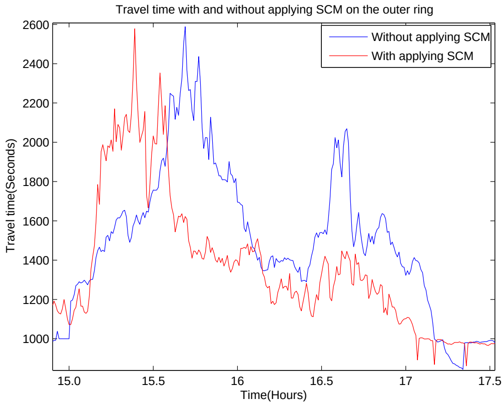


Figure 8.9: Travel time in the evening rush hour for the outer ring.

Table 8.3: ADR before and after applying SCM.

	Without applying SCM	With applying SCM	Improvement
Inner ring	26.89%	7.90%	18.99%
Outer ring	51.14%	37.01%	14.13%

much more travel time is spent on average, compared to the free flow travel time, and it is calculated with the following equation: $ADR = \frac{\Sigma_{i=1}^{n}(TT-TTF)/TTF}{n}$ where TT is the measured travel time and TTF is the free flow travel time. By comparing the ADR before and after applying the SCM, we can understand how much the traffic flow is improved by applying the SCM. The average delay rate during the evening rush hour can be seen from table 8.3. The ADR of the inner ring and the outer ring improved 18.99% and 14.13% respectively after applying the SCM. On average, the SCM improves the traffic flow on the ring road for about 16.56% in terms of the travel time delay.

8.3.4 Conclusions

In this thesis, we presented an approach to network control in a large network with different types of roads and more than one traffic management authority. The approach emerged out of the scenario-based approach, when the number of scenarios and the conflicts between scenarios grew out of hand, and resulted in a system called *Scenario Coordination Module*, a name which is still in use but is no longer fully appropriate. The system is in operation in the Amsterdam area. The approach can be combined with the multi-agent approach to network control, applied in urban networks, for groups of traffic signals, and at other sites. In this approach, the bottom-up control strategy is even more prominent. The performance evaluation shows that the SCM is a traffic network management control system with the capability to improve the network-wide traffic flow. In general, this combined approach has good prospects to become the standard way of doing network control in larger, mixed road networks.

8.4 Conclusions

In this chapter, two traffic control applications are presented. The HARS project is controlled by integrating Top-down and Bottom-up Control. It leads traffic situation to both a global and local optimum. Thus, the combined approach improves performance and efficiency of the traffic control system. However, in this project, the number of scenarios is limited. In the FileProof project, a system called SCM is used to control a large-scale road network with a large number of scenarios by combining the multi-agent approach to network control. Then the number of scenarios is distributed over the network. The performance evaluation shows that the SCM is a traffic network management control system with the capability to improve the network-wide traffic flow.

Chapter 9

Conclusions

This thesis mainly focuses on improving traffic flow for a large-scale road network by means of on-line distributed road traffic prediction and control. This thesis bridges the gaps between traffic engineering, control engineering and computer engineering. In this chapter, the main conclusions for each chapter will be drawn separately. Then, the overall conclusions of this thesis will be summarized. In the end, recommendations for future research will be presented.

9.1 Main conclusions for chapters

The main conclusions for each chapter can be drawn as follows:

Chapter 2 A literature survey is carried on to summarize the existing approaches of distributed traffic prediction and control. Based on the level of detail, traffic models can be classified into three types: microscopic, macroscopic and mesoscopic models. Among three types of models, macroscopic models are used in this thesis. Because they can accurately describe the average traffic dynamics of freeway traffic with a relatively less computation complexity. Then, the parallel simulation approaches are studied in order to simulate macroscopic models in a distributed way for a large-scale road network. The approaches are classified into four categories: Conservative, Optimistic, Relaxed synchronization and Mixed-mode. However, these approaches have a strong limitation on simulation concurrence which makes the simulation be determined by the slowest sub-simulator. Most existing distributed traffic application are based on these approaches, so that the applications inherit the drawbacks of the approaches. Recently, state matching approaches are applied in time-parallel simulation. Simulation can be performed concurrently by using the approaches. Furthermore, gluing algorithms are studied to solve the convergence problem for distributed simulation of physical systems and power grid systems. In this thesis, the approach is adapted to accommodate the traffic domain.

Traffic networked control approaches are classified into two categories: feedback control approaches and predictive control approaches. The main difference of the two approaches is that the latter approaches take the future traffic pattern into account. All the approaches are trying to solve the traffic problems individually. However, it is hard or even impossible to reach a system optimum. Furthermore, the centralized control from the traffic control center is no possible to perform any more. In this thesis, a multilevel control approach by combing a top-down and bottom-up approach is presented to overcome the problem.

Chapter 3 Two distributed architectures based on a new road network representation are presented to reduce the complexity of road networks and make the distributed prediction and control feasible. The new road network representation can be generated automatically by using a software generator. Furthermore, a distributed publish-subscribe middelware is introduced to further simplify the implementation of the distributed traffic prediction and control by hiding the complexity of low-level process communication. The same architectures and middleware are used for three purposes in later chapters: traffic monitoring, traffic simulation/prediction and traffic control. Both architectures can co-exist in a software application.

Chapter 4 Boundary points which do not have the knowledge of the traffic state upstream from those boundary points cannot use spatial information and traffic flow dynamics of upstream points to predict their states. Thus, an adaptive algorithm is presented to predict traffic demand at boundary points of a motorway network. The algorithm is based on an existing algorithm which was adjusted for the particular problem at hand and illustrated with the data of the ring road of Amsterdam. The results presented in this thesis show that the algorithm provides robust predictions with relatively small error of the next 30 minutes traffic demand both at motorways entries and at on-ramps. The boundary conditions provide more accurate inputs and initial guesses for the distributed prediction in chapter 5.

Chapter 5 An algorithm is presented to predict traffic flow in parallel for a large-scale motorway network. The algorithm is based on the state matching approach together the gluing algorithm. It leads the maximal simulation concurrence, at the mean time it guarantees the convergence of the simulation with a minimal number of iterations. It is illustrated by the performance of the algorithm for an academic network as well as for the Amsterdam ring network.

Chapter 6 The multilevel control framework presents the control at each level and how to synthesize, to design, and to evaluate a set of control laws to control a multilevel control system for a traffic network. The multilevel framework offers a scalable approach to network management as well as an approach to manage the many connections that Network Management systems have to make with existing systems in a network. The framework is the theoretical foundation of the two operational traffic network control systems in Chapter 8.

Chapter 7 Two operational traffic monitoring applications are presented. The systems provide the accurate, timely and reliable initial traffic states which are basic inputs for the distributed on-line traffic prediction in chapter 6. Both systems are based on the same distributed architecture which is described in Chapter 3. In case of sensors such as loops, cameras with license plate recognition, radars, GSM and GPS are available to collect traffic data, Treiber-Helbing-filter and Kalman filter are combined to complete missing data and estimate turn fraction. The results show that lost data is rebuilt efficiently with relatively small error. In case there are no sensors on motorways, Bluetooth detectors can be used to measure the average travel time. The results show that Bluetooth detectors are a cheap and reliable alternative sensors on freeways. The author of this thesis was in this project responsible for the system design, the part of the system implementation and the major part of the data analysis.

Chapter 8 Two operational traffic control systems are presented based on the multilevel control framework in chapter 6. Both systems are based on the same distributed architecture which is described in Chapter 3. The HARS project is controlled by integrating Top-down and Bottom-up Control. It leads traffic situation to both a global and local optimum. Thus, the combined approach improves performance and efficiency of the traffic control system. However, in this project, the number of scenarios is limited. In the FileProof project, a system called SCM is used to control a large-scale road network with a large number of scenarios by combining the multi-agent approach to network control. Then the number of scenarios is distributed over the network. The performance evaluation shows that the SCM is a traffic network management control system with the capability to improve the network-wide traffic flow. The author of this thesis was in this chapter responsible for the system design, the core part of the system implementation and the major part of the data analysis.

9.2 Summary of conclusions

Traffic congestion occurs more and more frequently due to a steady increase of traffic demand and to a much lower increase of road capacity. Thus, improving traffic flow by means of traffic control is the main focus in many countries. However, road traffic is one of the most complex system. Thus, it is also challenging to control the complex system.

9.2.1 Conclusions of challenges

The challenges in road traffic control are stated in Chapter 1 as follows:

On-line traffic prediction to develop a real-time prediction algorithm for on-line purposes, such as control scenarios evaluation.

Multilevel network control to network control for the coordination of local measures at different levels.

In this thesis, the above challenges are met by means of distributed prediction and control of traffic flow in a road network. The main conclusions of this thesis can be summarized as follows:

System simplification The complexity of road traffic is reduced by use of a new road network representation. Based on the representation, two distributed architectures are defined to be used by traffic monitoring in chapter 7, traffic prediction in chapter 4, 5 and traffic control in chapter 6, 8. Furthermore, in order to reduce the work load and prevent errors, a software generator is introduced to generate the new road network representation automatically. Last, a distributed publish-subscribe middleware called DSS datapool is introduced to further simplify the implementation of traffic monitoring, traffic prediction and traffic control.

Traffic demand prediction an adaptive prediction algorithm is presented for the inflows into the network in regular traffic situations based on stochastic control theory. The prediction algorithm is based on an adaptive prediction algorithm of T. Bohlin. The algorithm is designed and tested on traffic flow data of the ring road of Amsterdam. The results show that the algorithm provides robust predictions of traffic demand with relatively small errors for the next 30 min in a large-scale real-time environment.

Traffic prediction An algorithm is presented to predict traffic flow in parallel for a large-scale motorway network. The algorithm is based on the state matching approach together with the gluing algorithm. It leads to the maximal concurrence during simulation, at the mean time it guarantees the convergence of the simulation with a minimal number of iterations. It is illustrated by the performance of the algorithm for an academic network and for the Amsterdam ring network. It shows that the performance of the algorithm is much better than the performance of the centralized algorithm and the traditional algorithm.

Traffic monitoring To have accurate estimates of traffic states and traffic parameters (such as turn fraction) is essential for traffic prediction and traffic control. However, traffic data is not always available or cannot be measured directly. In this thesis, an approaching combing Treiber-Helbing-filter and Kalman filter is presented to supply missing (or error) data and estimate turn faction. In case there are no sensors on motorways, Bluetooth detectors can be used to measure the average travel time. The average travel time measured by Bluetooth detectors has been compared with travel time which measured by ALPR. The results show, that compared with ALPR, Bluetooth detectors can measure travel time as accurately but with relatively low cost and high reliability.

Traffic control Two traffic control systems are presented based on the multilevel traffic control framework which is proposed in chapter 6. The HARS project is controlled by integrating Top-down and Bottom-up Control. It leads the traffic situation to a global optimum. Thus, the combined approach improves performance and efficiency of the traffic control system. However, in this project, the number of scenarios is limited. In the FileProof project, a system called SCM is used to control a large-scale road network with a large number of scenarios by combining the multi-agent approach to network control. In this case the scenarios are distributed over the network. The performance evaluation shows that the SCM is a traffic network management control system with the capability to improve the network-wide traffic flow.

Computer engineering The author of this thesis carried out the deployment work including the system design, the (core) part of the system implementation and the major part of the data analysis of the traffic monitoring and traffic control systems.

9.2.2 Conclusions of three research questions

The three research questions stated in Chapter 1 have been answered in this thesis as follows:

Research question 1 The multilevel traffic control framework has been developed in chapter 6. The framework has been applied to two experimental case studies and two operational traffic control systems. It shows that it is a scalable framework, applicable to networks of any size.

Research question 2 Control and system theory as a tool has been applied to traffic prediction and control. The adaptive filtering and the coordinated parallel prediction algorithms have been developed to predict traffic demand and overall traffic in parallel, respectively. The parallel prediction is used to evaluate the performance of traffic control. However, the mathematical modeling of multilevel traffic control is still work in progress.

Research question 3 The distributed traffic prediction and control algorithms have been implemented in either the commercial middleware (DSS) or the experimental middleware (Agentscape). The distributed algorithms have been proved to be usable with different middlewares.

9.3 Recommendations and future research

Issues requiring further research for the distributed traffic prediction and control include:

Calibration and validation The default model parameters are used for the traffic models in this thesis. However, these parameter values may differ depending on the road conditions or on the weather conditions. Therefore, the parameters of the traffic model will be calibrated and validated on-line. Then the parallel simulation can accurately reflect the real traffic situation.

Different control measures Different traffic control measures are not considered in the parallel prediction algorithm in this thesis. However, in order to evaluate the impact of different traffic measures, they have to be taken into account in the parallel prediction. Then the parallel prediction algorithm can be used to accurately evaluate different control scenarios.

Detailed algorithms Detailed algorithms on control synthesize are not addressed in this thesis. Thus, control synthesis for multilevel control of traffic flow in a large-scale road network requires more research.

Deployment Implementation, testing, and evaluation of the on-line coordinated prediction algorithm in a practical application will be appreciated.

Bibliography

(2003) Alkmaar Brochure: Evaluation Dynamic Traffic Management, Tech. rep., Alkmaar Gemeente. 135

(2003) Handbook for sustainable traffic management, Tech. rep., Rijkswaterstaat, Rotterdam. 138

(2006) MaDAM, Tech. rep., Goudappel Goffeng B.V., maDAM is a traffic simulation model. 134

Aamodt, A., E. Plaza (1994) Case-based reasoning; foundational issues, methodological variations, and system approaches, *AI COMMUNICATIONS*, 7(1), pp. 39–59. 21

Adler, J. L., G. Satapathy, V. Manikonada, B. Bowles, V. J. Blue (2005) A multi-agent approach to cooperative traffic management and route guidance, in: *Transportation Research Part B: Methodological*, vol. 39, pp. 297–318. 23

Barcelo, J., J. L. Ferrer, F. Garcia, D., E. L. M., Saux (1998) Parallelization of microscopic traffic simulation for att systems analysis, in: *EQUILIBRIUM AND ADVANCED TRANSPORTATION MODELLING*. 19

Barcelo, J., L. Montero, L. Marques, C. Carmona (2010) Travel time forecasting and dynamic od estimation in freeways based on bluetooth traffic monitoring, in: *Proceedings of the 89th annual meeting of the Transport Research Board 2010*, pp. 1–17. 125, 126

Bemporad, A., M. Morari, V. Dua, E. N. Pistikopoulos (2002) The explicit linear quadratic regulator for constrained systems, *Automatica*, 38(1), pp. 3 – 20. 22

Bij, J. V. D., J. V. Schuppen (1983) Prediction of railway power demand, *Automatica*, 19(5), pp. 487 – 494. 42, 46

Bohlin, T. (1976) Four cases of identification of changing systems, in: Mehra, R. K., D. G. Lainiotis, eds., *System Identification Advances and Case Studies*, vol. 126 of *Mathematics in Science and Engineering*, Elsevier, pp. 441 – 518. 42

Boyd, S., L. Vandenberghe (2007) *Convex optimization*, (with corrections) ed., Cambridge University Press, Cambride, U.K. 173

Branston, D. (1976) Models of single lane time headway distributions, *Transportation Science*, 10(2), pp. 125 – 148. 15

Buckley, D. J. (1968) A semi-poisson model of traffic flow, *Transportation Science*, 2(2), pp. 107 – 133. 15

Camacho, E. F., C. A. Bordons (1997) *Model Predictive Control in the Process Industry*, Springer-Verlag New York, Inc., Secaucus, NJ, USA. 22

Cameron, G., G. Duncan (1996) Paramics-parallel microscopic simulation of road traffic, *The Journal of Supercomputing*, 10(1), pp. 25–53. 19

Chandler, R. E., R. Herman, E. W. Montroll (1958) Traffic dynamics: Studies in car following, *Operations Research*, 6(2), pp. pp. 165–184. 14

Chen, C.-T. (1984) *Linear system theory and design*, Holt, Rinehart and Winston, New York. 46

Chen, H.-F., P. R. Kumar, J. H. van Schuppen (1989) On Kalman filtering for conditionally Gaussian systems with random matrices, *Systems & Control Letters*, 13(5), pp. 397 – 404. 47

Cormen, T. H., C. E. Leiserson, R. L. Rivest, C. Stein (2001) *Introduction to Algorithms*, second edition ed., MIT Press and McGraw-Hill, section 24.3: Dijkstra's algorithm. 30

Corthout, R., C. M. J. Tampere, R. Frederix, L. H. Immers (2011) Marginal dynamic network loading for large-scale simulation-based applications, *Transportation Research Board 90th Annual Meeting*, pp. 1–18. 39

Cremer, M., H. Keller (1987) A new class of dynamic methods for the identification of origin-destination flows, *Transportation Research Part B: Methodological*, 21(2), pp. 117–132. 122

Cremer, M., J. Ludwig (1986) A fast simulation model for traffic flow on the basis of boolean operations, *Mathematics and Computers in Simulation*, 28(4), pp. 297 – 303. 14

Dai, W., J. Zhang, D. Zhang (2010) Parallel simulation of large-scale microscopic traffic networks, in: *Advanced Computer Control (ICACC), 2010 2nd International Conference on*, vol. 3, pp. 22 –28. 19

Davarynejad, M., Y. Wang, J. Vrancken, J. van den Berg (2011) Multi-phase time series models for motorway flow forecasting, in: *The 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, IEEE, pp. 2033–2038. 41

Davis, G., N. Nihan (1991) Nonparametric regression and short-term freeway traffic forecasting, *Journal of Transportation Engineering*, 117(2), pp. 178–188. 41

Ding, A., X. Zhao, L. Jiao (2002) Traffic flow time series prediction based on statistics learning theory, in: *The IEEE International Conference on Intelligent Transportation Systems*, IEEE, pp. 727–730. 41

Ding, Q., X. Wang, X. Zhang, Z. Sun (2011) Forecasting Traffic Volume with Space-Time ARIMA Model, *Advanced Materials Research*, 156-157, pp. 979–983. 41

Dinopoulou, V., C. Diakaki, M. Papageorgiou (2006) Applications of the urban traffic control strategy tuc, *European Journal of Operational Research*, 175(3), pp. 1652 – 1665. 22

Durbin, J., S. Koopman (2001) *Time Series Analysis by State Space Methods*, vol. 24 of *Oxford Statistical Science Series*, Oxford University Press, London, UK. 41

Eisenhauer, G., K. Schwan, F. Bustamante (2006) Publish-subscribe for high-performance computing, *Internet Computing, IEEE*, 10(1), pp. 40 – 47. 35

ESRI (1998) *ESRI Shapefile Technical Description*, ESRI. 26

ESRI (2007) ESRI Shapefile Technical Description, Tech. rep. 28

ESRI (2008) The website of ESRI company, http://www.esri.com/. 26

Faurre, P. L. (1976) Stochastic realization algorithms, in: Mehra, R. K., D. G. Lainiotis, eds., *System Identification Advances and Case Studies*, vol. 126 of *Mathematics in Science and Engineering*, Elsevier, pp. 1 – 25. 43, 44

Ferscha, A., J. Luthi (1995) Estimating rollback overhead for optimism control in time warp, in: *In Proceedings of the 28th Annual Simulation Symposium*, pp. 2–12. 17

Fiege, L., M. Cilia, G. Muhl, A. Buchmann (2006) 'publish-subscribe grows up: support for management, visibility control, and heterogeneity, *Internet Computing, IEEE*, 10(1), pp. 48 – 55. 35

Fujimoto, R. M. (2000) *Parallel and Distributed Simulation Systems*, John Wiley & Sons. 18

Gazis, D. C., R. Herman, R. W. Rothery (1961) Nonlinear follow-the-leader models of traffic flow, *Operations Research*, 9(4), pp. pp. 545–567. 14

Ghods, A., A. Rahimi-Kian (2008) A game theory approach to optimal coordinated ramp metering and variable speed limits, Yantai, China. 22, 23

Ghosh, B., B. Basu, M. O'Mahony (2009) Multivariate short-term traffic flow forecasting using time-series analysis, *Intelligent Transportation Systems, IEEE Transactions on*, 10(2), pp. 246 –254. 41

Gipps, P. (1981) A behavioural car-following model for computer simulation, *Transportation Research Part B: Methodological*, 15(2), pp. 105 – 111. 14

Goulart, P. J., E. C. Kerrigan, J. M. Maciejowski (2006) Optimization over state feedback policies for robust control with constraints, *Automatica*, 42(4), pp. 523 – 533. 22

Haghani, A., M. Hamedi, K. F. Sadabadi, S. Young, P. Tarnoff (2010) Freeway travel time ground truth data collection using bluetooth sensors, in: *Proceedings of the 89th annual meeting of the Transport Research Board 2010*, pp. 1–22. 125, 126, 130

Hamilton, A., B. Waterson, T. Cherrett, A. Robinson, I. Snell (2013) The evolution of urban traffic control: changing policy and technology, *Transportation Planning and Technology*, 36(1), pp. 24–43. 93

Hamnes, D. O., A. Tripathi (1994) Investigations in adaptive distributed simulation, in: *Proceedings of the eighth workshop on Parallel and distributed simulation*, PADS '94, ACM, New York, NY, USA, pp. 20–23. 17

Harvey, A. C. (1989) *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge University Press, Cambridge, UK. 41

Hegyi, A. (2004) *Model Predictive Control for the Integration of Traffic Control Measures*, Ph.D. thesis, Delft Univeristy of Technology. 23

Hegyi, A., B. De Schutter, S. Hoogendoorn, R. Babuska, H. Van Zuylen, H. Schuurman (2001) A fuzzy decision support system for traffic control centers, in: *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pp. 358–363. 21

Helbing, D., M. Schreckenberg (1999) Cellular automata simulating experimental properties of traffic flow, *Phys. Rev. E*, 59, pp. R2505–R2508. 14

Hong, W., Y. Dong, F. Zheng (2010) Forecasting urban traffic flow by SVR with continuous ACO, *Applied Mathematical Modelling*. 41

Hong, W., P. Pai, S. Yang, R. Theng (2006) Highway traffic forecasting by support vector regression model with tabu search algorithms, in: *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, IEEE, pp. 1617–1621. 41

Hoogendoorn, S., S. Hoogendoorn-Lanser, J. Van Kooten, S. Polderdijk (2011) Integrated network management: Toward an operational control method, in: *Transportation Research Board 90th Annual Meeting*, 11-2417. 94, 99

Hoogendoorn, S., B. D. Schutter, H. Schuurman (2003) Decision support in dynamic traffic management. real-time scenario evaluation, in: *European Journal of Transport and Infrastructure Research*, vol. 3 of *1*, pp. 21–38. 19

Hoogendoorn, S. P., P. H. Bovy (2001a) Generic gas-kinetic traffic systems modeling with applications to vehicular traffic flow, *Transportation Research Part B: Methodological*, 35(4), pp. 317 – 336. 15

Hoogendoorn, S. P., P. H. L. Bovy (2001b) State-of-the-art of vehicular traffic flow modelling, vol. 215, pp. 283–303. 14, 15

Hu, J., C. Zong, J. Song, Z. Zhang, J. Ren (2003) An applicable short-term traffic flow forecasting method based on chaotic theory, in: *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, vol. 1, pp. 608 – 613 vol.1. 41

Huang, Y. (2013) *Automated Simulation Model Generation*, Ph.D. thesis, Delft Univeristy of Technology. 6

Immers, B., C. Tampere, B. J. J. Stada (2008a) A multi-agent approach to dynamic traffic management, University of Leuven, Transportation Research Board, p. 24. 23

Immers, B., C. Tampère, J. Stada, B. Janssens (2008b) A multi-agent approach to dynamic traffic management, *University of Leuven, Transportation Research Board*, p. 24. 99

Jefferson, D. R. (1985) Virtual time, *ACM Trans. Program. Lang. Syst.*, 7(3), pp. 404–425. 16

J.H. van Schuppen, A. S., P. de Waal (1999) Simulation of a routing control algorithm for motorway networks, *Annex J DACCORD Deliverable D06.3*. 71

Johnston, C. M., A. T. Chronopoulos (1999) The parallelization of a highway traffic flow simulation, in: *Proceedings of the The 7th Symposium on the Frontiers of Massively Parallel Computation*, FRONTIERS ’99, IEEE Computer Society, Washington, DC, USA, pp. 192–. 20

Kai Nagel, Michael Schreckenberg (1992) A cellular automaton model for freeway traffic, *J. Phys. I France*, 2(12), pp. 2221–2229. 14

Katwijk, R. T. V. (2008) *Multi-Agent Look-Ahead Traffic-Adaptive Control*, Ph.D. thesis, Delft Univeristy of Technology. 23, 93

Kerner, B., S. Klenov (2006) Deterministic microscopic three-phase traffic flow models, *Journal of Physics A: Mathematical and General*, 39(8). 14

Kiesling, T., J. Luthi (2005) Towards time-parallel road traffic simulation, in: *Proceedings of the 19th Workshop on Principles of Advanced and Distributed Simulation*, PADS ’05, IEEE Computer Society, Washington, DC, USA, pp. 7–15. 18

Kiesling, T., S. Pohl (2004) Time-parallel simulation with approximative state matching, in: *Proceedings of the eighteenth workshop on Parallel and distributed simulation*, PADS ’04, ACM, New York, NY, USA, pp. 195–202. 18

Klefstad, R., Y. Zhang, M. Lai, R. Jayakrishnan, R. Lavanya (2005) A distributed, scalable, and synchronized framework for large-scale microscopic traffic simulation, in: *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pp. 813 – 818. 19, 69

Klein, U., T. Schulze, S. Straßburger (1998) Traffic simulation based on the high level architecture, in: *Proceedings of the 30th conference on Winter simulation*, IEEE Computer Society Press, pp. 1095–1104. 19

Kosmatopoulos, E., M. Papageorgiou, C. Bielefeldt, V. Dinopoulou, R. Morris, J. Mueck, A. Richards, F. Weichenmeier (2006) International comparative field evaluation of a traffic-responsive signal control strategy in three cities, *Transportation Research Part A: Policy and Practice*, 40(5), pp. 399 – 413. 22

Kossiakoff, A., W. N. Sweet, S. Seymour, S. M. Biemer (2011) *Systems engineering principles and practice*, vol. 83, John Wiley & Sons. 94

Kothare, M. V., V. Balakrishnan, M. Morari (1996) Robust constrained model predictive control using linear matrix inequalities, *Automatica*, 32(10), pp. 1361 – 1379. 22

Kotsialos, A., M. Papageorgiou (2004a) Efficiency and equity properties of freeway network-wide ramp metering with amoc, *Transportation Research Part C: Emerging Technologies*, 12(6), pp. 401 – 420. 21

Kotsialos, A., M. Papageorgiou (2004b) Motorway network traffic control systems, *European Journal of Operational Research*, 152(2), pp. 321 – 333, ¡ce:title¿New Technologies in Transportation Systems¡/ce:title¿. 21

Kotsialos, A., M. Papageorgiou, C. Diakaki, Y. Pavlis, F. Middelham (2002a) Traffic flow modeling of large-scale motorway networks using the macroscopic modeling tool metanet, *Intelligent Transportation Systems, IEEE Transactions on*, 3(4), pp. 282 – 292. 15

Kotsialos, A., M. Papageorgiou, M. Mangeas, H. Haj-Salem (2002b) Coordinated and integrated control of motorway networks via non-linear optimal control, *Transportation Research Part C: Emerging Technologies*, 10(1), pp. 65 – 84. 21

Kotsialos, A., M. Papageorgiou, F. Middelham (2005) Local and optimal coordinated ramp metering for freeway networks., *Journal of intelligent transportation systems : technology, planning and operations.*, 9(4), pp. 187–203. 22, 23

Krause, B., C. von Altrock (1997) A complete fuzzy logic control approach for existing traffic control systems, *In Mobility for Everyone, Proceedings of the 4th World Congress on Intelligent Transportation Systems*. 21

Kuhn, A., G. Eibl, T. Fasig (2012) Concept for an ‘‘intelligent‘‘traffic control network, in: *19th ITS World Congress*. 99

Kühne, R., R. Mahnke, I. Lubashevsky, J. Kaupužs (2002) Probabilistic description of traffic breakdowns, *Phys. Rev. E*, 65, p. 066125. 15

Kulcsar, B., I. Varga (2004) Simulation of turning rates in traffic systems, *In the Proceedings of IEEE European Simulation Symposium, Budapest, Hungary, October 17-20*, (LOG39). 122, 123

Kulcsar, B., I. Varga, B. Jozsef (2005) Constrained split rate estimation by moving horizon, *the 16th IFAC world congress*, pp. 1–6. 123

Kumar, P., P. Varaiya (1986) *Stochastic systems: Estimation, identification, and adaptive control*, Prentice Hall Inc., Englewood Cliffs, NJ. 43

Kwon, E., B. Choi, H. Park (2000) Distributed simulation of freeway traffic flow using personal computers, *Computer-Aided Civil and Infrastructure Engineering*, 15(3), pp. 167–175. 20

Lee, D.-H., P. Chandrasekar (2002) A framework for parallel traffic simulation using multiple instancing of a simulation program, *Journal of Intelligent Transportation Systems*, 7(3-4), pp. 279–294. 19

Lighthill, M. J., G. B. Whitham (1955) On kinematic waves: A theory of traffic flow on long crowded roads, *Proc. of the Royal Society, Series A*, 229, pp. 317–345. 15, 39, 71

Lin, S. (2011) *Efficient model predictive control for large-scale urban traffic networks*, Ph.D. thesis, Delft University of Technology. 22, 23

Lindquist, A., G. Picci (1985) Realization theory for multivariate stationary gaussian processes, *SIAM J. on control and Optimiz*, pp. 809–857. 44

Liu, H., H. V. Zuylen, H. V. Lint, M. Salomons (2006) Urban arterial travel time prediction with state-space neural networks and Kalman filters, in: *TRB 2006 Annual Meeting CD-ROM, Washington DC*. 41

Ljung, L. (1987) *System identification: Theory for the user*, Prentice Hall Inc., Englewood Cliffs, NJ. 45

Lubachevsky, B., A. Shwartz, A. Weiss (1989) Rollback sometimes works...if filtered, in: *Simulation Conference Proceedings, 1989. Winter*, pp. 630 –639. 17

Lubachevsky, B., B. Unger, D. Jefferson (1988) Bounded lag distributed discrete event simulation, in: *In Distributed Simulation*, pp. 182–192. 16, 17

Luenberger, D. (1969) *Optimization by vector space methods*, John Wiley & Sons, New York. 173, 174

Mahmassani, H. S., R. Jayakrishnan, R. Herman (1990) Network traffic flow theory: Microscopic simulation experiments on supercomputers, *Transportation Research Part A: General*, 24(2), pp. 149 – 162. 14

Malinovskiy, Y., Y.-J. Wu, Y. Wang, U.-K. Lee (2010) Field experiments on bluetooth-based travel time data collection, in: *Proceedings of the 89th annual meeting of the Transport Research Board 2010*. 125, 126

Marinica, N., R. Boel (2011) Particle filter for platoon based models of urban traffic, *ACC 2011*. 14

Mark, C., A. KauBer, M. Grein, H. Noltemeier (2004) *Computational Science and ITS Applications*, vol. 3044/2004 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, dynamically Changing Road Networks–Modelling and Visualization in Real Time. 26

Mayne, D., J. Rawlings, C. Rao, P. Scokaert (2000) Constrained model predictive control: Stability and optimality, *Automatica*, 36(6), pp. 789 – 814. 22

Mayne, D., M. Seron, S. Rakovic (2005) Robust model predictive control of constrained linear systems with bounded disturbances, *Automatica*, 41(2), pp. 219 – 224. 22

Messer, C. J. (1993) Advanced freeway system ramp metering strategies for texas, *Research report, Texas Transportation Institute*. 41

Messmer, A., M. Papageorgiou (1990) A macroscopic simulation program for motorway networks, *Traffic Engineering + Control*, 31, pp. 466–470. 39, 71

Nagel, K., M. Rickert (2001) Parallel implementation of the transims micro-simulation, *Parallel Computing*, 27(12), pp. 1611 – 1639, applications of parallel computing in transportation. 19, 69

Nagel, K., A. Schleicher (1994) Microscopic traffic modeling on parallel high performance computers, *PARALLEL COMPUTING*, 20, pp. 125–146. 14

Nakhla, M., R. Achar, D. Paul, N. Nakhla (2010) Parallel simulation of general electrical and mixed-domain circuits, *Data processing: structural design, modeling, simulation, and emulation modeling by mathematical expression*. 18

Nihan, N. L., G. A. Davis (1987) Recursive estimation of origin-destination matrices from input/output counts, *Transportation Research Part B: Methodological*, 21(2), pp. 149–163. 122

Noble, B. (1969) *Applied linear algebra*, Prentice-Hall, Englewood Cliffs, NJ. 179

NWB (2002) NWB(Nationaal Wegenbestand) report, Tech. rep. 26, 28

Okutani, I., Y. Stephanedes (1984) Dynamic prediction of traffic volume through Kalman filtering theory, *Transportation Research Part B: Methodological*, 18(1), pp. 1–11. 41

Ortega, J., W. Rheinboldt (1970) *Iterative solutions of nonlinear equations in several variables*, Academic Press, New York. 172

Panesar, K., R. Fujimoto (1997) Adaptive flow control in time warp, in: *Parallel and Distributed Simulation, 1997. Proceedings. 11th Workshop on*, pp. 108 –115. 17

Papamichail, I., M. Papageorgiou, V. Vong, J. Gaffney (2010a) Hero coordinated ramp metering implemented at the monash freeway, in: *Transportation Research Board Annual Meeting, 89th, 2010, Washington, DC, USA*, 10-0760. 114

Papamichail, I., M. Papageorgiou, V. Vong, J. Gaffney (2010b) Heuristic ramp-metering coordination strategy implemented at monash freeway, australia, *Transportation Research Record*, (2178), pp. 10–20, cited By (since 1996)11. 93

Paveri-Fontana, S. (1975) On boltzmann-like treatments for traffic flow: A critical review of the basic model and an alternative proposal for dilute traffic analysis, *Transportation Research*, 9(4), pp. 225 – 235. 15

Perumalla, K. (2006) Parallel and distributed simulation: Traditional techniques and recent advances, in: *Simulation Conference, 2006. WSC 06. Proceedings of the Winter*, pp. 84 –95. 16, 17, 18

Pooran, F., J. C. R. Martinez (2011) True adaptive control algorithms, a comparison of alternatives, in: *Proceedings of the 18th ITS World Congress*. 93

Potuzak, T. (2008) Distributed traffic simulation and the reduction of inter-process communication using traffic flow characteristics transfer, in: *Proceedings of the Tenth International Conference on Computer Modeling and Simulation*, UKSIM '08, IEEE Computer Society, Washington, DC, USA, pp. 525–530. 19

Potuzak, T. (2011) Comparison of road traffic network division based on microscopic and macroscopic simulation, in: *Proceedings of the 2011 UKSim 13th International Conference on Modelling and Simulation*, UKSIM '11, IEEE Computer Society, Washington, DC, USA, pp. 409–414. 69

Rathi, A. K. (1988) A control scheme for high traffic density sectors, *Transportation Research Part B: Methodological*, 22(2), pp. 81–101. 20, 133

Richards, P. I. (1956) Shock waves on the highway, *Operations Research*, 4(1), pp. 42–51. 15

Rickert, M., K. Nagel (2001) Dynamic traffic assignment on parallel computers in transims, *Future Gener. Comput. Syst.*, 17(5), pp. 637–648. 19

Ritchie, S. G. (1990) A knowledge-based decision support architecture for advanced traffic management, *Transportation Research Part A: General*, 24(1), pp. 27 – 37, Special Issue Expert Systems. 21

Robbins, B. A., V. M. Zavala (2011) Convergence analysis of a parallel newton scheme for dynamic power grid simulations, in: *Proceedings of the first international workshop on High performance computing, networking and analytics for the power grid*, HiPCNA-PG '11, ACM, New York, NY, USA, pp. 3–10. 18

Sakai, T., M. Nagao (1969) Simulation of traffic flows in a network, *Commun. ACM*, 12(6), pp. 311–318. 26

Siegelmann, H., E. Sontag (1991) Turing computability with neural nets, *Applied Mathematics Letters*, 4(6), pp. 77–80. 41

Smith, B., B. Williams, R. K. Oswald (2002) Comparison of parametric and nonparametric models for traffic flow forecasting, *Transportation Research Part C: Emerging Technologies*, 10(4), pp. 303–321. 39, 41

Smulders, S. (1986) Modelling and simulation of freeway traffic flow, *Report OS-R8615, Centre for Mathematics and Computer Science, Amsterdam*. 39, 71

Smulders, S. (1987a) Modelling and filtering of freeway traffic flow, *Proceedings of the 10th International Symposium on Transportation and Traffic Theory*, pp. 139–158. 71

Smulders, S. (1987b) Modelling and filtering of freeway traffic flow, Report os-r8706, Centre for Mathematics and Computer Science, Amsterdam. 72

Smulders, S. (1988) Filtering of freeway traffic flow, Report OS-R8806, Centrum voor Wiskunde en Informatica, Amsterdam. 72

Smulders, S. (1990) Control of freeway traffic flow by variable speed signs, *Transpn. Res.-B*, 24B, pp. 111–132. 72

Soares, M. S. (2010) *Architecture-Driven Integration of Modeling Languages for the Design of Software-Intensive Systems*, Ph.D. thesis, Delft Univeristy of Technology. 35

Söderström, T., P. Stoica (1989) *System identification*, Prentice Hall, New York. 45

Sontag, E. D. (1998) *Mathematical control theory: Deterministic finite dimensional system (2nd. Ed.)*, vol. 6 of *Graduate Text in Applied Mathematics*, Springer, New York. 46

Sotelo, M., J. Van Lint, U. Nunes, L. B. Vlacic, M. Chowdhury (2012) Introduction to the special issue on emergent cooperative technologies in intelligent transportation systems, *Intelligent Transportation Systems, IEEE Transactions on*, 13(1), pp. 1–5. 94

Stephanedes, Y. J., P. G. Michalopoulos, R. A. Plum (1981) Improved estimation of traffic flow for real-time control, *Transportation Research Record 795*, pp. 28–39. 41

Tanenbaum, A. S., M. V. Steen (2002) *Distributed Systems: Principles and Paradigms*, Prentice Hall. 33, 35

Thistle, J. (1999) Control synthesis for discrete event systems, in: Picci, G., D. Gilliam, eds., *Dynamical systems, control, coding, computer vision*, no. 25 in Progress in Systems and Control Theory, Birkhäuser Verlag, Boston, pp. 481–493.

Thulasidasan, S., S. Eidenbenz (2009) Accelerating traffic microsimulations: a parallel discrete-event queue-based approach for speed and scale, in: *Winter Simulation Conference*, WSC '09, Winter Simulation Conference, pp. 2457–2466. 19

Treiber, M., D. Helbing (2002) Reconstructing the spatio-temporal traffic dynamics from stationary detector data, *Cooperative Transport Dynamics*, (1:3.1-3.24). 35, 120

Treiber, M., A. Hennecke, D. Helbing (2000) Congested traffic states in empirical observations and microscopic simulations, *Phys. Rev. E*, 62, pp. 1805–1824. 14

van Lint, J., S. P. Hoogendoorn (2009) A robust and efficient method for fusing heterogenerous data from traffic sensors, *Computer Aided Infrastructure Engineering*. 120

van Lint, J., S. P. Hoogendoorn, M. Schreuder (2008) Fastlane- a new multi-class first order traffic flow model, *Transportation research record ISSN 0361-1981 CODEN TRREDM*. 15, 26, 39, 49, 71, 74, 87

van Schuppen, J. H., T. Villa, eds. (2014) *Coordination Control of Distributed Systems*, Springer. 115

van Wageningen-Kessels, F., J. van Lint, C. Vuik, S. Hoogendoorn (2012) Genealogy of traffic flow models, in: *Book of abstracts of the Highway Capacity and Quality of Service and Traffic Flow Theory and Characteristics Joint Symposium*. 14

van Wageningen-Kessels, F., B. van't Hof, S. P. Hoogendoorn, K. Vuik, H. van Lint (2011) Anisotropy in generic multi-class traffic flow models, *Transportmetrica*. 15

van Zuylen, H. J., L. Jie, L. Shoufeng (2010) The use of bleutooth scanners for travel time measurements. 130

Vapnik, V. (2000) *The nature of statistical learning theory*, Springer Verlag. 41

Vermijs, R. G. M. M., H. J. Papendrecht, L. R. F. Spelberg, W. J. Toetenel (1995) Short term forecasting of the level of service on a motorway network, by using a microscopic simulation model, Kerkrade, The Netherlands. 14

Vrancken, J., A. Brokx, R. Olsthoorn, A. Schreuders, M. Vale (2012) Dvm-exchange, the interoperability standard for network management systems, in: *19th ITS World Congress*. 93, 114

Vrancken, J., M. S. Soares (2007) Multi-level control of networks: the case of road traffic control, *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pp. 1741–1745. 26

Vrancken, J., Y. Wang, J. van Schuppen (2013) QHM: The quantitative hierarchical model for network-level traffic management, in: *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, pp. 223–230. 93

Vreeswijk, J., R. Blokpoel (2012) Uniform control targets and balanced strategies for traffic network management, in: *ITS World Congress*. 99

Wang, J., Z. dong Ma, G. M. Hulbert, B. Ci (2003) A gluing algorithm for distributed simulation of multibody systems, *Nonlinear Dynamics*, 34(1-2), pp. 159–188. 18

Wang, Y., M. Papageorgiou, A. Messmer (2008) Real-time freeway traffic state estimation based on extended kalman filter: Adaptive capabilities and real data testing, *Transportation Research Part A: Policy and Practice*, 42(10), pp. 1340–1358. 122

Wang, Y., J. van Schuppen, J. Vrancken (2014a) Prediction of traffic flow at the boundary of a motorway network, *Intelligent Transportation Systems, IEEE Transactions on*, 15(1), pp. 214–227. 39, 84, 89

Wang, Y., J. H. van Schuppen, J. Vrancken (2011a) Adaptive prediction of traffic flow into a motorway network, in: *Proc. ITS World congress*. 39, 84, 89

Wang, Y., J. H. van Schuppen, J. Vrancken (2013) Prediction of traffic demand at the boundary of a motorway network, Report, Faculty TBM, Delft University of Technology, Delft. 170

Wang, Y., J. H. van Schuppen, J. Vrancken (2014b) On-line distributed prediction of traffic flow in a large-scale road network, *Simulation Modelling Practice and Theory*, 47(0), pp. 276 – 303. 67

Wang, Y., J. H. van Schuppen, J. L. M. Vrancken (2012a) On-line distributed prediction of traffic flow in a large-scale traffic network, in: *Proceedings of the ITS World Congress*, pp. 1–8. 67

Wang, Y., J. Vrancken, F. Ottenhof, M. Vale (2012b) Next generation traffic control in the netherlands, *Traffic Engineering & Control*, 53(8). 99

Wang, Y., J. Vrancken, M. Soares (2009a) Intelligent network traffic control by integrating top-down and bottom-up control, in: *Control and Decision Conference, 2009. CCDC '09. Chinese*, pp. 1557 –1562. 13, 25, 39, 133

Wang, Y., J. Vrancken, M. S. Soares (2009b) Road network representation using Dijkstra's shortest path algorithm, iTS conference in Sweden 2009. 25, 86

Wang, Y., J. Vrancken, M. S. Soares (2010a) A distributed simulator for road network control, in: *Proceedings of IEEE International Conference on Networking, Sensing and Control*, pp. 20–25. 13, 39, 49, 50, 67, 71

Wang, Y., J. Vrancken, M. Valé and, M. Davarynejad (2010b) Integration of urban and freeway network control by using a scenario coordination module, in: *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp. 671 –676. 25, 39, 93, 133

Wang, Y., J. L. M. Vrancken, M. Davarynejad (2010c) Implementing scenarios coordination for road network traffic control, in: *Proceedings of the 2010 IEEE International Conference on Systems, Man and Cybernetics*, pp. 230–234. 25, 133

Wang, Y., J. L. M. Vrancken, Y. Ma (2012c) Evaluation of the scenario coordination module, a traffic network management control system, in: *Proceedings of the ITS World Congress*, pp. 1–8. 133

Wang, Y., J. L. M. Vrancken, F. Ottenhof, M. Valé (2011b) Next generation traffic control in the netherlands, in: *Proceedings of the ITS World Congress*, pp. 1–6. 133

Wang, Y., J. L. M. Vrancken, P. Seidel (2011c) Measure travel time by using bluetooth detectors on freeway, in: *Proceedings of the ITS World Congress*, pp. 1–5. 119

Wang, Y., J. L. M. Vrancken, M. S. Soares (2009c) Road network representation software generator, in: *12th IFAC Symposium on Control in Transportation Systems*, pp. 190–195. 25, 86

Wang, Y., Y. Yuan, J. L. M. Vrancken (2009d) Traffic parameter estimation on motorway networks by combination of filtering techniques, in: *Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics*, pp. 3658–3662. 119

Williams, B. (1999) *Modeling and forecasting vehicular traffic flow as a seasonal stochastic time series process*, Doctoral dissertation, Department of Civil Engineering, University of Virginia. 41

Williams, B., P. Durvasula, D. Brown (1998) Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models, *Transportation Research Record: Journal of the Transportation Research Board*, 1644(-1), pp. 132–141. 41

Wooldridge, M. (2009) *An introduction to multiagent systems*, John Wiley & Sons. 24

Wuest, B. A., D. Mioc (2007) Visualization and modeling of traffic congestion in urban environments, in: *10th AGILE International Conference on Geographic Information Science*, department of Geodesy and Geomatics, University of New Brunswick, Canada. 26

Xiao, Z., B. Unger, R. Simmonds, J. Cleary (1999) Scheduling critical channels in conservative parallel discrete event simulation, in: *Proceedings of the thirteenth workshop on Parallel and distributed simulation*, PADS '99, IEEE Computer Society, Washington, DC, USA, pp. 20–28. 16

Xiong, D. (1992) Traffic flow modeling with visualization tools, in: *GIS/LIS '92 Annual Conference and Exposition*, vol. 2, ACSM; ASPRS; AM/FM International; AG; URISA, San Jose, California, pp. 844–853, department of Geography, The Ohio State University. 26

Yuan, Y. (2013) *Langrangian multi-class traffic state estimation*, Ph.D. thesis, Delft University of Technology, Delft. 120

Zegeye, S. (2011) *Model-Based Traffic Control for Sustainable Mobility*, Ph.D. thesis, Delft University of Technology. 14, 22

Zhang, Y., Z. Ye (2008) Short-term traffic flow forecasting using fuzzy logic system methods, *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, 12(3), pp. 102–112. 41

Zhi, M. (2014) *Application of the Quantitative Hierarchical Model to Coordinated Ramp Metering*, Master's thesis, TU Delft, Delft University of Technology. 114

Zijpp, N. J. V. D. (1997) Dynamic origin-destination matrix estimation from traffic counts and automated vehicle identification data, *Transportation Research Record: Journal of the Transportation Research Board*, pp. 87–94. 122

Appendices

Appendix A

Adaptive Prediction Algorithm[1]

A.1 MATLAB source code

The core MATLAB source code of the adaptive prediction algorithm is as follows:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%week model
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%inital value
x=A4_1100_week1_10; % initial states
y=A4_1100_week2_10; % measurement data
q0=0.1; % inital value for Riccati equation
q1=0.1; % noise for states
q2=0.1; % noise for measurement
q=q0;
%calculate k
k=q/(q+q2);
%update x
%loop for t=t0...t1
 clear xnew;
 for t = 1:length(x)
xnew(t)=x(t)+k*(y(t)-x(t));
end
%update q Riccati equation
qnew=q+q1-q*q/(q+q2);
%calculate new q
qnewy=qnew+q2;
%update x and q
x=xnew;
```

[1] Code published courtesy of Trinité Automation B.V., Uithoorn.

```
q=qnew;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%short term model
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
v5=0.009; % state noise
qh=0.9; %measurement noise
He=[-0.1;-0.1;-0.1;0.1;0.1;0.1]; % passed measurements
ehat=0;
%Ln,Mn,Nn
LnTemp=[zeros(1,3);eye(2),zeros(2,1)];
Ln=[LnTemp,zeros(3,3);zeros(3,3),LnTemp];
Mn=zeros(3*2,1);
Mn(1)=-1;
Nn=zeros(3*2,1);
Nn(4)=1;
%predict the new values of parameters
%calculate ke
ke=qe*He/(He'*qe*He+qh);
%calculate penew and qenew
% parameters
penew=pe+ke*(e-He'*pe);
% variance of error in parameters
qenew=qe+v5*eye(6)-qe*He*He'*qe/(He'*qe*He+q);
%predict the next ehat
Henew=Ln*He+Mn*ehat+Nn*e;
%calculate ehatnew
ehatnew=Henew'*penew;
%update
He=Henew;
pe=penew;
ehat=ehatnew;
%The first 10 mins prediction
i=1;
yhat(i)=x(t)*(1+ehatnew);
%update
Hehnew=Henew;
%The next 20 mins and 30 mins prediction
for i=2:3
Hehnew=Ln*Hehnew+(Mn+Nn)*ehatnew;
ehatnew=Hehnew'*penew
eh(i,t)=ehatnew;
yhat(i)=x(t+i-1)*(1+ehatnew);
end
```

Appendix B

On-line Distributed Traffic Prediction Algorithm

B.1 Algorithm for the Prediction of Traffic Flow in a Road Network

The overall algorithm is summarized. The algorithm includes statements for the computation of the fixpoint equation which have been discussed in Section B.2.

Algorithm B.1.1 Algorithm of prediction of traffic flow in a road network by coordinated distributed computations. *Consider the prediction systems defined above.*
Data. *The systems of the subnetworks. The initial time* $t_0 \in \mathbb{N}$, *the duration of one discrete-time step* $t_s \in \mathbb{R}_+$, *the time step between the successive predictions* $t_r \in \mathbb{Z}_+$, *the prediction horizon* $t_p \in \mathbb{Z}_+$, *and the terminal time* $t_1 \in \mathbb{Z}_+$. *(Example,* $t_0 = 0$, $t_s = 10$ sec, $t_r = 60$, $t_p = 180$ *(corresponds to 30 minutes), and* $t_1 = 1080$ *(corresponds to 3 hours).) The maximum number of iterations of the convergence* k_{max}. *The acceptable value of the approximation criterion, the tolerance,* $\varepsilon_{tol} \in (0,\infty) \subset \mathbb{R}$.
Initialization. *Set the prediction of the motorway inputs* $\hat{u}_{net,j}(t_0+1 : t_0+t_p|t_0)$ *at time* $t_0 \in \mathbb{Z}_+$ *equal to the values of the same vector measured last week for the prediction horizon.*

Carry out the computations:

1. *For* $t = t_0,\ t_0+t_r,\ t_0+2t_r, \ldots, t_1$, *do,*

   (a) *At all subnetworks* $j \in J$ *compute asynchronously in parallel:*

      i. *Read the estimate of the current state of the filter systems* $\hat{x}_{net,j}(t)$. *The filter system is not part of this algorithm.*

      ii. *Read the subnetwork traffic inflows which are not coming from motorways thus are coming from on-ramps,* $\hat{u}_{in}(t+1 : t+t_p|t)$. *These* *predictions are produced by another part of the prediction system and may be based on the method described in Wang et al. (2013).*

(b) *For $k = 1,\ 2, \ldots, k_{max}$ if $\varepsilon_{net}^{(k)} > \varepsilon_{tol}$ compute:*

i. *At all subnetworks $j \in J$ compute asynchronously in parallel,*

A. *Compute by simulation,*

$$s = 1, \ldots, t_p,$$

$$\begin{aligned}\hat{x}_{net,j}^{(k+1)}(t+s+1|t) &= f(\hat{x}_{net,j}^{k+1)}(t+s|t), \hat{u}_{net,j,m}^{(k)}(t+s|t),\\ &\quad \hat{u}_{net,j,in}(t+s|t)),\end{aligned}$$

$$\hat{x}_{net,j}^{(k+1)}(t|t) = \hat{x}_{net,j}(t), \tag{B.1}$$

$$\hat{z}_{net,j}^{(k+1)}(t+s|t) = h_{net,j}(\hat{x}^{(k+1)}(t+s)), \tag{B.2}$$

$$\hat{z}_{net,j}^{(k+1)}(t+1:t+t_p|t) = \begin{pmatrix} \hat{z}_{net,j}^{(k+1)}(t+1|t) \\ \hat{z}_{net,j}^{(k+1)}(t+2|t) \\ \vdots \\ \hat{z}_{net,j}^{(k+1)}(t+t_p|t) \end{pmatrix} \in \mathbb{R}^{t_p p_{net,j}} \tag{B.3}$$

B. *Send from Subnetwork j to each of its downstream subnetworks indexed by $J_{out}(j) \subset J$, the predicted outflow of motorway traffic of Subnetwork j, $\hat{z}_{net,j}^{(k+1)}(t+1:t+t_p|t)$.*

C. *If appropriate, receive from one of the upstream subnetworks indexed by $J_{in}(j) \subset J$ their prediction of motorway outflows which are inflows of Subnetwork j.*

D. *Compute the new motorway inflow of Subnetwork $j \in J$ based on the successive approximation method, see Section B.2, by insertion of the newly computed values at the appropriate places. The algorithm applies a supply-demand step developed in the reference . This step allows for the propagation of a traffic congestion from a subnetwork to an upstream or a downstream network.*

$$\hat{u}_{net,j,m,i}^{(k+1)}(t+1|t) = \begin{cases} \hat{z}_{net,i}^{(k+1)}(t+1|t), & \textit{if } i \in J_{in}(j), \\ \hat{u}_{net,j,m,i}(t+1|t), & \textit{else}, \end{cases} \tag{B.4}$$

$\hat{u}_{net,j,m,i}^{(k+1)}(t+2|t), \ \ldots, \hat{u}_{net,j,m,i}^{(k+1)}(t+t_p|t),$
correspondingly, hence obtain,
$\hat{u}_{net,j,m}^{(k+1)}(t+1:t+t_p|t).$

E. *Compute the local value of the approximation criterion of Subnetwork j.*

$$\varepsilon_{net,j}^{(k+1)} = \|\hat{u}_{net,j,m}^{(k+1)}(t+1:t+t_p|t) - \hat{u}_{net,j,m}^{(k)}(t+1:t+t_p|t)\|_2.$$

F. *Send from Subnetwork j to the network prediction system the current value of the approximation error* $\varepsilon_{net,j}^{(k+1)}$.

ii. *At the network prediction system compute the new value of the approximation error and check the obtained value against the tolerance, according to,*

$$(\varepsilon_{net}^{(k+1)})^2 = \sum_{j=1}^{n_{net}} (\varepsilon_{net,j}^{(k+1)} c_j)^2 / \sum_{j=1}^{n_{net}} c_j, \; c_j = \dim \varepsilon_{net,j}. \quad \text{(B.5)}$$

$$\varepsilon_{net}^{(k+1)} \leq \varepsilon_{tol}. \quad \text{(B.6)}$$

The iteration has converged or not depending on whether $\varepsilon_{net}^{(k+1)} \leq \varepsilon_{tol}$ *or the converse holds respectively.*

iii. *The prediction system at the network level does, depending on the case:*

A. *if* $\varepsilon_{net}^{(k+1)} > \varepsilon_{tol}$ *and if* $k < k_{max}$ *then the iteration continues;*

B. *if* $\varepsilon_{net}^{(k+1)} > \varepsilon_{tol}$ *and if* $k \geq k_{max}$ *then the network prediction system sends to all subnetworks a message with the information that convergence was not reached in* k_{max} *steps after which the algorithm stops;*

C. *if* $\varepsilon_{net}^{(k+1)} \leq \varepsilon_{tol}$ *then the prediction system sends to the prediction systems of all subnetworks a message with the information that convergence was reached in* $k \in \mathbb{Z}_+$ *steps and then the algorithm leaves the iteration step of the algorithm;*

iv. *Each subnetwork prediction system asynchronously and in parallel does:*

A. *Receives from the network prediction system the information about the convergence or the nonconvergence of the iteration; continue the algorithm in case of convergence; stop the algorithm in case of nonconvergence;*

B. *Communicates to its local station: the predicted traffic flows of the state* $\hat{x}_{net,j}(t_1 : t+t_p|t)$*, of the off-ramp flows, and of the motorway outflows at the boundary points of the network* $\hat{z}_{net,j}(t+1:t+t_p|t)$ *and* $\hat{z}_{net,j,out}(t+1:t+t_p|t)$*;*

v. *The subnetwork computes the initial prediction of the motorway inflows for the next time step of the prediction,*

$$\hat{u}_{net,j,m}^{(0)}(EXP|t+t_r) = \begin{cases} \hat{u}_{net,j,m}^{(k+1)}(EXP|t), \\ \quad \text{if } EXP = t+t_r+1 : t+t_p, \\ u_{net,j,m}(EXP), \\ \quad \text{if } EXP = t+t_p+1 : t+t_p+t_r. \end{cases} \quad \text{(B.7)}$$

where $u_{net,j,m}(.)$ denotes the traffic flow data of last week or an application of the adaptive predictor to the past predictions.

B.2 Solutions of a fixpoint equation

In Section 5.3, a fixpoint equation was derived for the traffic inflows into a subnetwork. In this section it is show how to compute a solution to such a fixpoint equation.

Problem B.2.1 *Consider the fixpoint equation for the vector of predicted inflows. Determine the vector of inflows which is a solution of the fixpoint equation (5.15).*

The computation of the solution of a fixpoint equation is a known topic of mathematics, see for example the book Ortega & Rheinboldt (1970). The following methods can be used for the fixpoint equation of the prediction of traffic flow: (1) a successive approximation method; and (2) a recursive method for a short horizon. These methods are described in the following subsections.

The successive approximation methods require the selection of an approximation criterion. The approximation criterion will be taken to be L_2-norm of a vector $v \in \mathbb{R}^k$ as $\|v\|_2 = [\sum_{i=1}^{k} v_i^2]^{1/2}$. Other norms could be used.

In case that after one iteration the approximation criterion of a subnetwork is already zero then there is no need for that subnetwork to recompute from the predicted motorway inflows the corresponding predicted motorway outflows. This then applies also for subnetworks downstream from that with the zero error. Note that in the case that a subnetwork has only motorway traffic inflows from outside the network the error will be zero after one iteration. The zero approximation error of a subnetwork can be used to further reduce the computational complexity but this is not discussed further.

Of interest to the problem of this chapter is the distributed character of the computation for the approximation of the solution of the fixpoint equation. Distinguish the following aspects of the fixpoint computation:

1. The recursive computation of successive approximations of the fixpoint equation;
2. The computation of the value of the approximation criterion.

These items are discussed below.

Consider first the recursive computation of the successive approximations of the fixpoint equation. The computations are in principle carried out at the subnetwork level where the subsystem is available. Any subnetwork needs to have available the predictions of the inflows into the subnetwork. These predictions for the prediction horizon of traffic inflows consists of (1) the prediction of the outflow of upstream subnetworks, say $\hat{z}_{net,j}(t+1:t+t_p|t)$; and (2) the prediction of traffic inflow at the boundary of the network, say $\hat{u}_{net,j,in}(t+1:t+t_p|t)$. The latter predictions do not change during the computations for a particular time moment. The first predictions have to be communicated by the upstream subnetwork to a downstream network. Thus, the recursive computation requires only the communication of computed predictions of traffic flow of a subnetwork to a downstream subnetwork. In this section an algorithm is provided for the solution of the fixpoint equation defined in the previous section.

B.2.1 Successive Approximation method

Problem B.2.2 *Consider the fixpoint equation, repeated from equation (5.15),*

$$v = f_{fp}(t,v), \text{ for the vector } v = \hat{u}_{net,m}(t+1:t+t_p|t) \in \mathbb{R}^{m_{net,m}}. \tag{B.8}$$

Formulate an algorithm to compute an approximation of the solution of this equation which allows a coordinated-distributed computation.

The coordinated-distributed computation is defined to consists of synchronous parallel computations by the subnetworks and a check at the network level after each step of the iteration or after a few steps.

Below are successively discussed the centralized computation and the distributed computation of the solution of the fixpoint equation.

Consider then first the centralized computation in which the solution of the fixpoint equation is computed at the network location. The main algorithms used to compute an approximation of the fixpoint equation are:

- The *successive approximation method* which consists of the formulas

  $$v(k+1) = f_{fp}(t,v(k)),\ v(0) = v_0 \in \mathbb{R}^{n_v},\ n_v = m_{net,m},\ \{v(k),\ k \in \mathbb{N}\}. \tag{B.9}$$

  In practice one stops the sequence of iterations, $\{v(k), k \in \mathbb{N}\}$, after a finite number of steps when a convergence criterion has dropped below a tolerance level.

- The Newton method for a fixpoint equation, see (Luenberger, 1969, Sec. 10.3).

- The *steepest descent method*, see (Luenberger, 1969, Sec. 10.4).

The Newton method is also stated in (Boyd & Vandenberghe, 2007, Section 9.5).

For any algorithm to compute the solution of the fixpoint equation, the following questions have to be investigated:

1. Does there exist a solution $\hat{u}(t+1:t+t_p|t) \in \mathbb{R}^{n_v}$ of the equation (5.15)?

2. If there exists a solution, is the solution unique?

3. Does the considered algorithm produce an approximation sequence which converges to a solution of the equation?

4. Can the convergence rate of the recursion be made fast by an appropriate choice of the recursion?

There exists an unique solution of the fixpoint equation if the function $f_{fp}(t,.)$ satisfies the condition of a contraction map or if diagonal dominance holds, see (Luenberger, 1969, Sec. 10.3). The convergence of the successive approximation methods is not the fastest possible. The Newton method results in general in a faster convergence than the successive approximation method at the expense of more computations per iteration.

A centralized computation for the predictions of the motorway inputs converges for many examples as simulation indicates. There is a structural property that supports this experience, the fixpoint equation possesses a dissipation property, the motorway inputs have an effect on the outputs of the same subnetwork only after the number of steps required for the fastest car to reach from the entry point the exit point of the subnetwork.

Consider next the distributed computation of the solution of a fixpoint equation. The reader should think of a road network with two or more subnetworks. The distributed computation requires a specification of the algorithm in regard to the levels of the network and an analysis of the computational complexity.

The distributed computation of the successive approximation method is proposed in this chapter. Algorithm B.1.1 specifies the distributed computations for the network and the subnetwork levels. At any subnetwork the computations can be carried out for that subnetwork. Next it is required that each subnetwork communicates its temporal local solution to downstream subnetworks if any. After receipt of temporal local solutions of upstream networks, the subnetwork can compute the value of the local approximation criterion. At the network level then only the value of the approximation criterion of the full network has to be computed. The computational complexity and the communication complexity are stated in Proposition 5.4.1.

In section B.3 a mathematical analysis and a numerical example is provided for the computation of the solution of the fixpoint equation when the fixpoint function is an affine map. In that case the answers to the above questions of existence and of uniqueness of a solution and the convergence rates follow directly from the eigenvalues of the matrix of the linear function. The eigenvalues belong all to the interior of the stability domain of the spectrum hence convergence is established.

In examples of network prediction problems the convergence seems always to occur. The number of iterations required for a tolerance of 10^{-k} with $k = 2$ is about 1 or 2 iterations.

The algorithm of the distributed Newton method has a higher computational complexity. An approximation of the inverse of the first derivative of the fixpoint function has to be computed in a distributive way. The authors have neither carried out computations nor a mathematical analysis for a distributed Newton algorithm also because the experiences with the distributed successive approximation algorithm are quite positive.

The conclusion of this subsection is that the distributed successive approximation algorithm for the approximation of the solution of the fixpoint equation is quite effective and converges sufficiently fast for the application considered.

B.2.2 Recursive Method

The authors have after an investigation decided not to implement this method. Therefore it is shortly summarized. In this subsection the recursive method is presented to achieve coordination of the traffic predictions of subnetworks at the network level. The formalization is preceded by an example.

Example B.2.3 *Consider the ring network consisting of two subnetworks. Note that the effect of a prediction for the motorway input at the point where a particular subnetwork starts will have an effect on the predictions of the motorway outflow at the* same location *only if there was sufficient time for traffic to drive around the full ring. Thus, if the ring has a length of 9 km as in this example and if the travel speed is at most 120 km/h, than the travel time of the traffic flow to travel around the full loop is* $9 \times 60/120 = 4,5$ min $= 270$ sec. *In* 240 sec $= 4$ min *or* 24 *steps of* 10 sec *each, then there is no effect of the motorway input on the motorway output at the same location.*

Therefore a recursion can be set up to recursively compute the predictions over the horizon. Denote by t_d for delay time, the minimal number of time steps of 10 sec each for the traffic to travel around the ring. Take the value of the recursion step $t_r \in \mathbb{Z}_+$ to be such that $t_r < t_d$. In the above example, $t_d = 270$ s. while $t_r = 240$ s. The computations are then,

$$\begin{aligned}
& && \hat{u}_{net,m,h}(t+1 : t+t_r|t), \\
& && \textit{assumed available from the initialization} \\
& && \textit{or from the previous time step,} \\
\hat{z}_{net,m,h}(t+1 : t+t_r|t) &= && f_{net,m,h}(\hat{x}_{net}(t), \hat{u}_{net,m,h}(t+1 : t+t_r|t)), \\
\hat{u}_{net,m,h}(t+t_r+1 : t+2t_r|t) &= && \hat{z}_{net,m,h}(t+t_r+1 : t+2t_r|t), \\
\hat{z}_{net,m,h}(t+k\times t_r+1 : t+(k+1)\times t_r|t) &= && f_{net,m,h}(\hat{x}_{net}(t), \\
& && \hat{u}_{net,m,h}(t+k\times t_r+1 : t+(k+1)\times t_r|t)), \\
& && \textit{etc.}
\end{aligned}$$

The number of recursion steps depends on the length of the prediction horizon and the length of the duration t_r chosen.

By the above recursive method one can compute the output flows of the motorway network over the prediction horizon.

The principle of the recursive method to achieve coordination of the predictions of the subnetworks is thus to determine the duration of the recursion and then to carry out the predictions in a recursive manner. The computation of the recursion duration depends on the network considered. If the network has one or more rings, the recursion duration is strictly less than the shortest travel time around the ring. If the network has no rings then choose the recursion duration about half the length of the prediction horizon to take care of the fact that at time t there are no prediction for the last 10 minutes of the 30 minute horizon in the example.

The notation and the recursions follow.

Algorithm B.2.4 The recursive coordination of predictions of traffic flow in a road network.
Data. $t_0 \in \mathbb{N}$ *the initial time,* $t_1 \in \mathbb{Z}_+$ *the terminal time,* $t_s \in \mathbb{R}_+$ *the duration of a time step in sec,* $t_q \in \mathbb{Z}_+$ *the time step at which predictions are to be computed,* $t_p \in \mathbb{Z}_+$ *the duration of the prediction horizon in steps,* $t_d \in \mathbb{R}$ *the minimal time needed to drive around the shortest ring network. (Example.* $t_0 = 0$, $t_1 = 1080\,\text{sec}$, $t_s = 10\,\text{sec}$, $t_q = 60$, $t_p = 180$, $t_d = 900\,\text{sec}$. *)*

1. *Compute* $t_r \in \mathbb{Z}_+$ *as the maximal number such that* $(t_r + 2) \times t_s < t_d$ *where the two extra steps in* $t_s + 2$ *are a precautionary measure,* $t_r = \min\{t_r, t_p\}$ *to guarantee that the number or recursion steps is smaller than the prediction horizon. Define* $k_{max} = \lceil t_p / t_r \rceil \in \mathbb{Z}_+$.

2. *For* $t = 0, t_s, 2t_s, \ldots, t_1$ *do,*

   (a) *For* $k = 0, 1, \ldots, k_{max} - 1$ *do,*

      i.

$$\begin{aligned}
&\hat{u}^{(1)}_{net,m}(t + kt_r + 1 : t + kt_r + t_r | t) \\
&\qquad \textit{select arbitrary, for example equal to last weeks traffic inflow,} \\
&\hat{z}^{(1)}_{net,m}(t + kt_r + 1 : t + kt_r + t_r | t) \\
&\quad = \; f_{net,j}(\hat{x}(t + kt_r | t), \\
&\qquad \hat{u}^{(1)}_{net,m}(t + kt_r + 1 : t + kt_r + t_r | t), \hat{u}_{in,j}(t + kt_r + 1 : t + kt_r + t_r | t)), \\
&\hat{u}^{(2)}_{net,m}(t + kt_r + 1 : t + kt_r + t_r | t) \\
&\quad = \; \hat{z}^{(1)}_{net,m}(t + kt_r + 1 : t + kt_r + t_r | t), \\
&\hat{z}^{(2)}_{net,m}(t + kt_r + 1 : t + kt_r + t_r | t) \\
&\quad = \; f_{net,j}(\hat{x}(t + kt_r | t), \hat{u}^{(2)}_{net,m}(t + kt_r + 1 : t + kt_r + t_r | t) \\
&\qquad \hat{u}_{in,j}(t + kt_r + 1 : t + kt_r + t_r | t)), \qquad \text{(B.10)} \\
&\hat{x}(t + t_r | t) \\
&\qquad \textit{computed along with } \hat{z}^{(2)}_{net,m}(t + kt_r + 1 : t + kt_r + t_r | t). \qquad \text{(B.11)}
\end{aligned}$$

In Algorithm B.2.4 the step of Equation (B.10) is not explicitly needed, it is a by product of Equation (B.11) for the computation of the predicted state $\hat{x}(t+t_r|t)$. The advantage of the recursive method is that the computations are carried out in a recursive method and that *no* iterations of the predictions over the full horizon are required. This is in contrast with the fixpoint method where iterations of the predictions over the full horizon are required.

Above the recursive method is described for the case of a centralized computation. The reader can now easily formulate the algorithm in case the computations are carried out at all subnetworks and coordinated at the network level.

The authors do not favor the recursive method for the fixpoint equation. The successive approximation method compares the current prediction with that of the previous step and then improves by another iteration if the tolerance is not met.

B.3 Fixpoint Method – Linear Case

In this section the fixed-point method is analysed for the simple ring network with a linear system. This subsection is a continuation of the Examples 5.3.1, 5.3.5, and 5.3.11.

B.3.1 The system

The focus of the fixed-point method is in this subsection restricted to a ring network with two parts, labelled Subnetwork 1 and Subnetwork 2. The dynamics of the system and of the output equation are affine maps. Within this scope one can prove that the fixpoint equation has an unique solution if a condition on a matrix is met. In Section 5.5 computations will be shown for the case of a ring network with two parts and linear equations.

Definition B.3.1 *Define of Subnetwork 1 the linear system by the equations,*

$$\begin{aligned} x_1(t+1) &= A_1x_1(t)+B_{1,m}u_{1,m}(t)+B_{1,in}u_{1,in}(t),\ x_1(t_0)=x_{1,0}, \\ z_{1,m}(t) &= C_1x_1(t). \end{aligned}$$

For simplicity it is assumed that there is no inflow into the last section before the point at which Part 1 of the network ends.

The solution of the state and of the output is then,

$$\begin{aligned} x_1(t) &= A_1^{t-t_0}x_1(t_0)+\sum_{s=t_0}^{t-1}A_1^{t-s-1}B_{1,m}u_{1,m}(s)+\sum_{s=t_0}^{t-1}A_1^{t-s-1}B_{1,in}u_{1,in}(s), \\ z_{1,m}(t) &= C_1A_1^{t-t_0}x_1(t_0)+\sum_{s=t_0}^{t-1}C_1A_1^{t-s-1}B_{1,m}u_{1,m}(s)+\sum_{s=t_0}^{t-1}C_1A_1^{t-s-1}B_{1,in}u_{1,in}(s). \end{aligned}$$

Definition B.3.2 *Define the* horizon-prediction system of the Part 1 Subnetwork. *Define the variables, (1) the state ($x_{1,0}(t|t)$) at time $t \in T$; (2) the predicted inflow on the prediction horizon from the main motorway ($u_{1,m}(t+1:t+t_p|t)$), and (3) the predicted inflow on the prediction horizon from on-ramps ($u_{1,in}(t+1:t+t_p|t)$). The representation follows.*

$$
\begin{aligned}
\hat{z}_{1,m}(t_0:t_1-1|t) &= \begin{pmatrix} \hat{z}_{1,m}(t+1|t) \\ \hat{z}_{1,m}(t+2|t) \\ \vdots \\ \hat{z}_{1,m}(t+t_p|t) \end{pmatrix}, \ \hat{u}_{1,m}(t+1:t+t_p|t), \ \hat{u}_{1,in}(t+1:t+t_p|t), \\
H(A_1,B_{1,m},C_1) &= \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ C_1B_{1,m} & 0 & 0 & \dots & 0 \\ C_1A_1B_{1,m} & C_1B_{1,m} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ C_1A_1^{t_1-1}B_{1,m} & C_1A_1^{t_1-2}B_{1,m} & \dots & C_1B_{1,m} & 0 \end{pmatrix}, \\
\text{obsmat}(A_1,C_1)_{t_1} &= \begin{pmatrix} C_1 \\ C_1A_1 \\ C_1A_1^2 \\ \vdots \\ C_1A_1^{t_1-1} \end{pmatrix}, \\
\hat{z}_{1,m}(t+1:t+t_p|t) &= \text{obsmat}(A_1,C_1)\hat{x}_1(t|t) + H(A_1,B_{1,m},C_1)\hat{u}_{1,m}(t+1:t+t_p|t) + \\
&\quad + H(A_1,B_{1,in},C_1)\hat{u}_{1,in}(t+1:t+t_p|t).
\end{aligned}
$$

B.3.2 The Ring Network

Consider next the simple ring network consisting of two parts referred to as Part 1 and Part 2. The corresponding relations for the trajectories of these parts and their interconnections are then described by the equations,

$$
\begin{aligned}
\hat{z}_{1,m}(t+1:t+t_p|t) &= \text{obsmat}(A_1,C_1)\hat{x}_1(t) + H(A_1,B_{1,m},C_1)\hat{u}_{1,m}(t+1:t+t_p|t) + \\
&\quad + H(A_1,B_{1,in},C_1)\hat{u}_{1,in}(t+1:t+t_p|t), \\
\hat{z}_{2,m}(t+1:t+t_p|t) &= H(A_2,B_{2,m},C_2)\hat{u}_{2,m}(t+1:t+t_p|t) + \text{obsmat}(A_2,C_2)\hat{x}_2(t) + \\
&\quad + H(A_2,B_{2,in},C_2)\hat{u}_{2,in}(t+1:t+t_p|t), \\
\hat{u}_{1,m}(t+1:t+t_p|t) &= \hat{z}_{2,m}(t+1:t+t_p|t), \\
\hat{u}_{2,m}(t+1:t+t_p|t) &= \hat{z}_{1,m}(t+1:t+t_p|t), \ \forall\, t \in T.
\end{aligned}
$$

Definition B.3.3 *Define the* fixpoint equation *for the input vectors of the motorway traffic inflows in case of a ring network with two parts and with linear dynamics by the* *following equation and definitions,*

$$\begin{aligned}
\begin{pmatrix} \hat{u}_{1,m} \\ \hat{u}_{2,m} \end{pmatrix} &= +\begin{pmatrix} 0 & H(A_2,B_{2,m},C_2) \\ H(A_1,B_{1,m},C_1) & 0 \end{pmatrix}\begin{pmatrix} \hat{u}_{1,m} \\ \hat{u}_{2,m} \end{pmatrix}+ \\
&+\begin{pmatrix} 0 & \text{obsmat}(A_2,C_2) \\ \text{obsmat}(A_1,C_1) & 0 \end{pmatrix}\begin{pmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \end{pmatrix}+ \\
&+\begin{pmatrix} 0 & H(A_2,B_{2,in},C_2) \\ H(A_1,B_{1,in},C_1) & 0 \end{pmatrix}\begin{pmatrix} \hat{u}_{1,in} \\ \hat{u}_{2,in} \end{pmatrix} \qquad \text{(B.12)} \\
\hat{u}_m &= A_m\hat{u}_m + b_m, \ \textit{or,} \\
b_m &= F_m\hat{u}_m, \ \textit{where,} \\
A_m &= \begin{pmatrix} 0 & H(A_2,B_{2,m},C_2) \\ H(A_1,B_{1,m},C_1) & 0 \end{pmatrix}, \\
F_m &= I - A_m, \\
b_m &= \begin{pmatrix} 0 & \text{obsmat}(A_2,C_2) \\ \text{obsmat}(A_1,C_1) & 0 \end{pmatrix}\begin{pmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \end{pmatrix}+ \\
&+\begin{pmatrix} 0 & H(A_2,B_{2,in},C_2) \\ H(A_1,B_{1,in},C_1) & 0 \end{pmatrix}\begin{pmatrix} \hat{u}_{1,in}(t+1:t+t_p|t) \\ \hat{u}_{2,in}(t+1:t+t_p|t) \end{pmatrix} \\
\hat{u}_m &= \begin{pmatrix} \hat{u}_{1,m}(t+1:t+t_p|t) \\ \hat{u}_{2,m}(t+1:t+t_p|t) \end{pmatrix}.
\end{aligned}$$

The definition above leads to the conclusion that one must analyse the fixpoint equation (B.12).

Proposition B.3.4 *Consider the fixpoint equation (B.12).*

(a) If $\text{rank}(F_m) = \text{rank}(F_m\ b_m)$ *and this number equals the number of elements of the vector* $\hat{u}_m$ *(the number of unknowns) then there exists an unique solution of the fixpoint equation. Denote then the solution by* $\hat{u}_m^* \in \mathbb{R}^{m_{mp}}$ *hence* $F_m\hat{u}_m^* = b_m$.

(b) If there exists an unique solution and if the eigenvalues of the matrix A_m *are in the interior of the unit circle in the set of the complex numbers,* $\mathbb{C}$, *then the sequence* $\{\hat{u}_m^{(k)} \in \mathbb{R}^{m_{mp}},\ k \in \mathbb{Z}_+\}$ *produced by the successive approximation*

$$\hat{u}_m^{(k+1)} = A_m\hat{u}_m^{(k)} + b_m, \qquad \text{(B.13)}$$

converges to the solution, $\lim_{k\to\infty}\hat{u}_m^{(k)} = \hat{u}_m^*$. *The convergence speed is determined by the eigenvalues of the matrix* A_m.

Proof (a) This is a standard result of linear algebra, see for example (Noble, 1969, Th. 3.6.b).

(b) The recursion is well defined and may be interpreted as the simulation of the system modeling the road network. Denote the difference between the sequence produced by the recursion and the solution by,

$$\begin{aligned} \Delta u^{(k)} &= \hat{u}_m^{(k)} - \hat{u}_m^*, && \text{(B.14)} \\ \Delta u^{(k+1)} &= A_m \hat{u}_m^{(k)} + b_m - A_m \hat{u}_m^* - b_m = A_m \Delta \hat{u}^{(k)}. && \text{(B.15)} \end{aligned}$$

It is then a standard result of linear algebra or of linear difference equations that the eigenvalues of A_m in the interior of the unit circle in the complex numbers implies that the sequence converges to the zero vector, hence $\lim_{k\to\infty} \hat{u}_m^{(k)} = \hat{u}_m^*$. It is clear from the above displayed formula that the convergence speed of $\Delta\hat{u}^{(k)}$ is determined by the eigenvalues of the matrix A_m.

Proposition B.3.5 *Consider the matrices,*

$$\begin{aligned} A_m &= \begin{pmatrix} 0 & H(A_2, B_{2,m}, C_2) \\ H(A_1, B_{1,m}, C_1) & 0 \end{pmatrix}, && \text{(B.16)} \\ H_1 &= H(A_1, B_{1,m}, C_1),\ H_2 = H(A_2, B_{2,m}, C_2). && \text{(B.17)} \end{aligned}$$

Any eigenvalue $\lambda(A_m)$ of the matrix A_m equals either the positive or the negative square root of an eigenvalue of the matrix H_1H_2. The eigenvalues and the eigenvectors of the matrix A_m can be computed by the steps,

$$\begin{aligned} & \textit{solve for the eigenvalues } \lambda(H_1H_2) \textit{ of the matrix } H_1H_2, \\ & \textit{solve the following equation for the vector } v_2 \in \mathbb{C}^k, \\ H_1H_2v_2 &= \lambda(H_1H_2)v_2, \textit{ and set,} && \text{(B.18)} \\ v_1 &= +/-\lambda(H_1H_2)^{-1}H_2v_2,\ \lambda(A_m) = +/-\lambda(H_1H_2)^{1/2}, && \text{(B.19)} \\ v &= \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}, \textit{ then,} && \text{(B.20)} \\ A_m v &= \lambda(A_m)v. && \text{(B.21)} \end{aligned}$$

Proof Note the calculations,

$$\begin{aligned} \lambda(A_m)v &= A_m v = \begin{pmatrix} 0 & H_2 \\ H_1 & 0 \end{pmatrix} v, && \text{(B.22)} \\ &\Leftrightarrow\ H_2v_2 = \lambda(A_m)v_1,\ H_1v_1 = \lambda(A_m)v_2, \\ &\quad \Rightarrow\ H_1H_2v_2 = \lambda(A_M)H_1v_1 = \lambda(A_m)^2 v_2, \\ &\text{from which } \lambda(A_m) \text{ and } v_2 \text{ follow,} \\ &\lambda(A_m)v_1 = H_2v_2,\ \Rightarrow v_1 = \lambda(A_m)^{-1}H_2v_2. && \text{(B.23)} \end{aligned}$$

B.3.3 Linear Case – Numerical Example

There follows a numerical example of the simple ring network of the previous subsection with linear dynamics and output equation.

Consider the simple ring network, see Example 5.3.1 and 5.3.5. The ring is partitioned into two parts labelled Subnetwork 1 and the Subnetwork 2 as shown in Fig. 5.1 respectively as the left-half and as the right-half of the ring. Subnetwork 1 consists of four sections or cells with one on-ramp and one off-ramp. Subnetwork 2 consists of five cells with one on-ramp and two off-ramps.

Denote the indices of the road network by,

$$= \{1,2\}, \tag{B.24}$$

$$(1) = \{1,2,3,4\},\ (2) = \{1,2,3,4,5\}, \tag{B.25}$$

$${}_{m}(1) = \{1\},\ {}_{on}(1) = \{3\},\ {}_{of}(1) = \{2,4\}, \tag{B.26}$$

$${}_{m}(2) = \{1\},\ {}_{on}(2) = \{3\},\ {}_{of}(2) = \{2,4,5\}. \tag{B.27}$$

The dynamics of the traffic flow is as defined in the linear system of Def. B.3.1. The values of the parameters of this system of the two subnetworks are,

$$\begin{aligned}
A_1 &= \begin{pmatrix} 0.5 & 0 & 0 & 0 \\ 0.5 & 0.4 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0.4 \end{pmatrix},\ B_{1,m} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix},\ B_{1,on} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \\
C_1 &= \begin{pmatrix} 0 & 0 & 0 & 0.4 \end{pmatrix}, \\
A_2 &= \begin{pmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0.5 & 0.4 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0.4 & 0 \\ 0 & 0 & 0 & 0.5 & 0.4 \end{pmatrix},\ B_{2,m} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},\ B_{2,on} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \\
C_2 &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0.4 \end{pmatrix}.
\end{aligned}$$

Of interest are the eigenvalues of the matrices A_1 and A_2. These are $\operatorname{spec}(A_1) = \{0.4, 0.4, 0.5, 0.5\}$ and $\operatorname{spec}(A_2) = \{0.4, 0.4, 0.4, 0.5, 0.5\}$. The eigenvalues can be read off from the diagonal of the matrices because the structure of the matrices consists of a diagonal and the first subdiagonal. A computation shows that the matrix A_m has eigenvalues strictly inside the unit circle of the complex numbers hence that the conditions for the existence of a unique fixed-point of Equation (B.12) are satisfied.

B.4 Source code of an agent

The calculations of the traffic model is performed in the function simulateNetwork(). Road network is read from a database in Matlab and stored to a Matlab file, then the Matlab file is read in an agent. Road network means how many segments and how many meters per segment for each subnetwork. The core agent source code is as follows:

```
public class DefaultAgent extends Agent
{
        //@Override
        //define variables and their initial values
        private int TimeStep = 5;
        private int CriticalDensity = 70;
        private int MaxiFlow = 2000;
        private int RampOutflow = 30;
        private int InflowBoundary = 500;
        private int OutflowBounday =2000;
        private int VelocityMax=120;
        private int VelocityCrit=80;
        private int DensityMax=200;

        @Override
        public void run()
        {
        //define agent object ID
        String objectId = ”Subnetwork1”;

        //regester itself to dirtory service
        startWithName(objectId);

        //Read a subnetwork from a matlab data file
        MatFileReader matfilereader
        = new MatFileReader(”Subnetwork1.mat” );
        double[][] data
        = ((MLDouble)matfilereader.getMLArray
                        ( objectId )).getArray();

        //Simulate the subnetwork for next 30 minutes
        for (int i = 0; i < 360; i++)
        {
            // simulate for each time step
            simulateNetwork(data);
        }
```

```
    //send demand, supply to downstream and upstream
    sendMessageTo(DownstreamList, demand);
    sendMessageTo(UpstreamList, supply);

    //receive message from downstream and upstream
    receiveMessageFrom();

    //use min supply demand scheme to update outflow
    InflowBoundaryUpdated
    = Math.min(inflow, supplyOfFirstCell);
    OutflowBoundayUpdated
    = Math.min(supplyFromUpstream, outflow);

    //check consistence: re-do simulation
    //if the inflow and outflow are not consistent
    if(InflowBoundaryUpdated != InflowBoundary
    && OutflowBoundayUpdated != OutflowBounday)
    {
            //re-do simulation for next 30 minutes
            for (int i = 0; i < 360; i++)
            {
            // simulate for each time step
            simulateNetwork(data);
            }
    }
    }

private boolean simulateNetwork (double[][] data)
{
   // for all cells
   for(int j = 0;j < data[0].length; j++)
   {
    // calculate demand and supply for each cell
    CalculateDemandSupply();

    //calculate outflow based on
    //minimal demand and supply scheme
    CalculateOutflow();

    // cell length
    CellLength = data[0][j];
```

```
        //update density by conservation equation
        UpdateDensity();

        //Update velocity by fundamental diagram
        UpdateVelocity();

        //Update flow: flow = density * velocity
        UpdateFlow();
       }

       //store output: density, velocity and flow
       StoreOutput();
     }
}
```

Summary

Traffic becomes more and more congested due to the growing mismatch between supply and demand for road capacity. As a consequence, improving traffic flow by means of traffic control is a key focus in many countries. However, road traffic is one of the most complex systems and therefore also complex to control. In this thesis, a distributed solution for traffic prediction and control is proposed to reduce this complexity and achieve fast computation of traffic predictions and control for the purpose of on-line (real-time) usage in a large-scale motorway network.

The main challenge of this thesis is to achieve multilevel network control by using an on-line traffic prediction algorithm. A goal for this algorithm is, for instance, control scenario evaluation. The on-line traffic prediction algorithm works as an evaluation tool to guarantee the best possible performance of the multilevel network control algorithm. This algorithm allows traffic to be controlled by just local measures when possible, and coordination of local measures at different geographic scopes in case of traffic problems that cannot be handled locally.

The main contributions to this challenge by this thesis can be summarized as follows:

Complexity reduction The complexity of road traffic is reduced by introducing a new road network representation. Based on this representation, two distributed system architectures are defined that are applied to systems for traffic monitoring, prediction and control. Converting a given network into this representation is labour-intensive and error prone. In order to reduce the work load and to prevent errors, a software generator is explained to generate the new road network representation automatically. Finally, a distributed Publish-Subscribe middelware, called the DSS Datapool, is introduced to further simplify the implementation of systems for traffic monitoring, prediction and control.

Traffic demand prediction An adaptive prediction algorithm is presented for the inflows into the network under regular traffic circumstances, based on stochastic control theory. The prediction algorithm is derived from an adaptive prediction algorithm by T. Bohlin. The algorithm is designed and tested with traffic flow data from the belt road of Amsterdam. The results show that the algorithm provides robust predictions of traffic demand with relatively small errors for a time horizon of 30 minutes in a large-scale, real-time environment.

Traffic prediction An algorithm is presented to predict traffic flow, using parallel computation, for a large-scale motorway network. The algorithm is based on a state matching approach (for adjacent subareas) together with a gluing algorithm. It leads to maximal simulation parallelism, while guaranteeing the convergence of the simulation in a minimal number of iterations. The performance of this algorithm is much better than either the performance of the centralized algorithm or of the traditional algorithm, for both a simple, synthetic network and for the Amsterdam ring network.

Traffic monitoring Accurate estimates of traffic states and traffic parameters (such as turn fractions), are essential for traffic prediction and traffic control. However, traffic data is not always available or cannot be measured directly. In this thesis, an approach combining the Treiber-Helbing-filter and a Kalman filter is presented to supply missing (or replace erroneous) data and estimate turn fractions. In case there are no other sensors, such as induction loops, on motorways, Bluetooth detectors can be used to measure average travel time. The average travel time measured by Bluetooth detectors has been compared with travel times measured by Automated License Plate Recognition (ALPR). The results show that Bluetooth detectors can measure travel times with comparable accuracy but with relatively low equipment costs and high reliability.

Traffic control A multilevel traffic control framework is proposed to control and coordinate traffic at different geographic scopes. Two operational traffic control systems are presented based on this framework in order to demonstrate the framework in practice. The HARS system, operational on Alkmaar's belt road, is a two-level system using top-down and bottom-up control. It realizes both a globally and locally optimal traffic state. Thus, the combined approach improves performance and efficiency of the traffic control system. However, in this system, the number of scenarios is limited. In the FileProof project in Amsterdam, a three-level system called SCM (Scenario Coordination Module) is used to control a large-scale road network with agent-based control and a huge number of control scenarios. Different scenarios for different parts of the network can be combined. The performance evaluation of this system has shown that SCM is a network management system with the capability to improve network-wide traffic flow.

Samenvatting

Het verkeer raakt steeds verder verstopt door de groeiende kloof tussen vraag en aanbod in wegcapaciteit. Vandaar dat het verbeteren van de doorstroming door middel van verkeersmanagement in veel landen veel aandacht krijgt. Maar, het verkeer behoort tot de meest complexe systemen. Het is dan ook een complex probleem om dit complexe systeem te regelen. Dit proefschrift beschrijft een gedistribueerde aanpak voor het voorspellen en regelen van verkeer waarmee complexiteit wordt gereduceerd en waarmee de rekensnelheid bereikt kan worden die nodig is voor on-line (dat wil zeggen: tijdens de operatie) gebruik van verkeersvoorspellingen in een grootschalig netwerk.

De belangrijkste uitdaging voor dit proefschrift was het realiseren van een meerlaagse netwerksturing door gebruik van een on-line verkeersvoorspellingsalgoritme. Een belangrijke toepassing van dit algoritme is het on-line evalueren van regelscenarios. Het algoritme werkt als een on-line evaluatietool om te zorgen dat de meerlaagse verkeersregeling zo goed mogelijk presteert. Deze verkeersregeling coördineert lokale regelingen zodat verkeersproblemen op een geschikte geografische schaal worden opgelost: lokaal wat lokaal kan, en voor een geschikt groter gebied als dat nodig is.

De belangrijkste bijdragen aan bovengenoemde uitdaging en conclusies van dit proefschrift kunnen als volgt worden samengevat.

Complexiteitsreductie De complexiteit van wegverkeer wordt verminderd door de gebruik van een nieuwe netwerkrepresentatie. Met deze representatie worden twee gedistribueerde architecturen gedefinieerd en gebruikt voor systemen voor verkeersmetingen, verkeersvoorspelling en verkeersregeling. Voor het arbeidsintensieve en fout-gevoelige werk van het genereren van deze netwerkrepresentatie is een software-hulpmiddel ontwikkeld en beschreven in dit proefschrift. Tenslotte wordt een Publiceer-en-Abonneer communicatie-middenlaag (beter bekend als Publish-Subscribe Middleware) geïntroduceerd om implementatie van deze systemen voor meting, voorspelling en regeling verder te vereenvoudigen.

Verkeersvraagvoorspelling Een adaptief voorspellingsalgoritme, gebaseerd op stochastische regeltheorie, wordt beschreven voor het voorspellen van de instroom in een netwerk bij reguliere verkeersomstandigheden. Dit algoritme is afgeleid van een adaptief algoritme van T. Bohlin. Het algoritme is ontworpen en getest op verkeersdata van de ringweg om Amsterdam. De resultaten laten zien dat het algoritme real-time, robuuste voorspellingen levert, met betrekkelijk kleine fouten, voor een tijdshorizon van 30 minuten, in een grootschalig netwerk.

Verkeersvoorspelling Een algoritme wordt beschreven om, met geparallelliseerd rekenwerk, de verkeersstroom te voorspellen in een grootschalig autosnelwegennetwerk. Dit algoritme combineert het aan elkaar passen van de verkeerstoestand in aangrenzende deelgebieden met een plak-algoritme. Het leidt tot maximale parallellisering van het rekenwerk, en garandeert tegelijkertijd de convergentie naar een consistente uitkomst in een minimaal aantal herhalingen. De prestaties van dit algoritme zijn aanzienlijk beter dan die van een gecentraliseerd algoritme, zowel op een simpel, kunstmatig netwerk, als op het netwerk rond de Amsterdamse ringweg.

Verkeersmeting De beschikbaarheid van nauwkeurige informatie over de verkeerstoestand en verkeersparameters, zoals afslagfracties, in een netwerk, is essentieel voor verkeersvoorspelling en verkeersregeling. Verkeersdata is echter niet altijd beschikbaar of kan niet direct gemeten worden. In dit proefschrift wordt beschreven hoe ontbrekende data aangevuld kan worden door middel van een combinatie van een Treiber-Helbing filter en een Kalman filter, en hoe hiermee foutmarges of afslagfracties geschat kunnen worden. Wanneer er geen andere sensoren, zoals detectielussen, op een snelweg aanwezig zijn, kunnen Bluetooth-detectors gebruikt worden om gemiddelde reistijd te berekenen. De waarden, gemeten met Bluetooth-detectors, is vergeleken met dezelfde waarden gemeten met camera's en kentekenherkenning. De resultaten laten zien dat met Bluetooth-detectors reistijden gemeten worden met een vergelijkbare nauwkeurigheid, maar met betrekkelijk lage kosten voor de apparatuur en toch hoge betrouwbaarheid.

Verkeersregeling Er wordt een voorstel gedaan voor een meerlaags regelraamwerk om verkeer te regelen en te coördineren op verschillende geografische schalen. Twee operationele verkeersregelsystemen, gebaseerd op dit raamwerk, worden beschreven die laten zien hoe dit raamwerk in de praktijk functioneert. Het tweelaags HARS-systeem, op de ringweg van Alkmaar, maakt gebruik van top-down en bottom-up sturing. Het realiseert daarmee een lokaal en globaal optimale verkeersregeling. Deze combinatie verbetert prestaties en efficiëntie van het verkeersregelsysteem. In dit systeem is het aantal scenario's evenwel beperkt. In het FileProof project wordt een drielaags systeem, genaamd SCM (Scenario-coördinatiemodule), gebruikt dat grote aantallen scenario's kan toepassen door een agent-gebaseerde aanpak te combineren met netwerkmanagement. Daarbij kunnen verschillende scenario's op verschillende delen van het netwerk gecombineerd worden. De prestatie-evaluatie van dit systeem laat zien dat SCM een netwerkmanagementsysteem is met de mogelijkheid om netwerk-breed de verkeersstroom te verbeteren.

概述 (Summary in Chinese)

由于道路容量的提供和需求之间存在越来越大的差别，交通状况变的越来越拥挤。因此，利用智能交通控制的方式来提高交通流量已经成为许多国家的关注点。然而，道路交通是一个非常复杂的系统。因此，道路交通的控制也是非常复杂的。我的论文研究课题是利用一个分布式的解决方案来进行交通预测和交通控制，从而来减少交通系统的复杂性和提高交通预测和控制的快速性，以达到对一个大型高速公路网络的实时控制目的。

我的论文研究课题的主要挑战性是在用一个实时的分布式交通预测算法来达到多层次的交通网络控制。 这个交通预测算法的主要目的是评估交通控制预案的性能。这个交通预测算法作为一个评估的工具来保证多层次交通网络控制方案是最优化的。这个网络控制方案允许在可能的情况下仅仅用局部控制器来控制交通，当交通问题不能被局部控制的时候用协调各个局部控制器的方式来控制。

我的论文研究课题的挑战性主要被概括如下：

减少复杂性 道路交通的复杂性被一个新的道路网络表示方式来减少。基于这个表示方式，两个系统架构被定义并应用于交通监控，交通预测和交通控制系统上。把一个交通网络转变成这个表示方式是需要大量人力并且不可避免的出现很多人为错误。 为了减少人力和避免错误，一个软件生成器被用来自动生成这个网络表示方式。最后一个叫DSS数据池的分布式发布-订阅中间层被应用进一步简化交通监控，交通预测和交通控制系统的开发。

交通需求预测 在正常交通状况下，一个自适应的预测算法被应用在交通网络的的边缘上来预测进入这个网络的流量。这个自适应的预测算法在T. Bohlin的自适应的预测算法上演变而来的。这个算法被应用在Amsterdam环城高速公路上采集的交通流量数据。这个测试的结果证明这个算法能够在实时的环境下给大型交通网络提供稳定的而且误差很小交通需求的流量预测。

交通预测 一个应用平行同步计算方式的算法被用来预测大型高速公路的交通流量。 这个算法是在对应状态的方式和连接算法的基础上演变而来的。这个同步算法使得仿真同步最大化的同时保证仿真在最少的循环下完成聚焦。结果证明这个算法的性能比起其他算法的性能都要好很多。

交通监控 精确的估算交通状态和交通参数是交通预测和控制的至关点。然而，交通数据不是总是被测量的或者不能被直接测量。 一个结合Treiber-Helbing-filter和Kalman filter的算法来补充没有的数据和估算转弯率。如果没有传统的交通数据源，蓝牙接收器可以用来测量平均行驶时间。蓝牙测量的平均行使时间与ALPR测量的平均行使时间进行了比较，结果证明两者之间的测量没有太大的区别，但是蓝牙设备更加便宜和可靠。

交通控制 一个多层的交通控制算法被用来控制和协调在不同地理位子的交通。基于这个算法，两个交通控制系统已经被应用在实际中。 运行在Alkmaar环形路中的HARS系统是一个2层的结合自上而下和自下而上的控制系统。这个系统能够识别局部和全局的最优交通状态。因此，这个算法能够提高交通控制系统的性能和有效性。然而，这个系统能控制的交通预案是有限的。在Amsterdam的FileProof项目中， 一个叫SCM的3层交通控制系统被用来控制大型的交通网络，其中有基于Agent的控制和非常巨大的交通预案控制。不同的区域的交通预案能被结合起来。结果证明这个系统能够有效的提高整个交通网络的流量。

TRAIL Thesis Series

The following list contains the most recent dissertations in the TRAIL Thesis Series. For an overview of more than 100 titles see the TRAIL website: www.rsTRAIL.nl.

The TRAIL Thesis Series is a series of the Netherlands TRAIL Research School on transport, infrastructure and logistics.

Wang, Y., On-line Distributed Prediction and Control for a Large-scale Traffic Network, T2015/6, March 2014, TRAIL Thesis Series, the Netherlands

Vreeswijk, J.D., The Dynamics of User Perception, Decision Making and Route Choice, T2015/5, February 2015, TRAIL Thesis Series, the Netherlands

Lu, R., The Effects of Information and Communication Technologies on Accessibility, T2015/4, February 2015, TRAIL Thesis Series, the Netherlands

Ramos, G. de, Dynamic Route Choice Modelling of the Effects of Travel Information using RP Data, T2015/3, February 2015, TRAIL Thesis Series, the Netherlands

Sierzchula, W.S., Development and Early Adoption of Electric Vehicles: Understanding the tempest, T2015/2, January 2015, TRAIL Thesis Series, the Netherlands

Vianen, T. van, Simulation-integrated Design of Dry Bulk Terminals, T2015/1, January 2015, TRAIL Thesis Series, the Netherlands

Risto, M., Cooperative In-Vehicle Advice: A study into drivers‘‘ability and willingness to follow tactical driver advice, T2014/10, December 2014, TRAIL Thesis Series, the Netherlands

Djukic, T., Dynamic OD Demand Estimation and Prediction for Dynamic Traffic Management, T2014/9, November 2014, TRAIL Thesis Series, the Netherlands

Chen, C., Task Complexity and Time Pressure: Impacts on activity-travel choices, T2014/8, November 2014, TRAIL Thesis Series, the Netherlands

Wang, Y., Optimal Trajectory Planning and Train Scheduling for Railway Systems, T2014/7, November 2014, TRAIL Thesis Series, the Netherlands

Wang, M., Generic Model Predictive Control Framework for Advanced Driver Assistance Systems, T2014/6, October 2014, TRAIL Thesis Series, the Netherlands

Kecman, P., Models for Predictive Railway Traffic Management, T2014/5, October 2014, TRAIL Thesis Series, the Netherlands

Davarynejad, M., Deploying Evolutionary Metaheuristics for Global Optimization, T2014/4, June 2014, TRAIL Thesis Series, the Netherlands

Li, J., Characteristics of Chinese Driver Behavior, T2014/3, June 2014, TRAIL Thesis Series, the Netherlands

Mouter, N., Cost-Benefit Analysis in Practice: A study of the way Cost-Benefit Analysis is perceived by key actors in the Dutch appraisal practice for spatial-infrastructure projects, T2014/2, June 2014, TRAIL Thesis Series, the Netherlands

Ohazulike, A., Road Pricing mechanism: A game theoretic and multi-level approach, T2014/1, January 2014, TRAIL Thesis Series, the Netherlands

Cranenburgh, S. van, Vacation Travel Behaviour in a Very Different Future, T2013/12, November 2013, TRAIL Thesis Series, the Netherlands

Samsura, D.A.A., Games and the City: Applying game-theoretical approaches to land and property development analysis, T2013/11, November 2013, TRAIL Thesis Series, the Netherlands

Huijts, N., Sustainable Energy Technology Acceptance: A psychological perspective, T2013/10, September 2013, TRAIL Thesis Series, the Netherlands

Zhang, Mo, A Freight Transport Model for Integrated Network, Service, and Policy Design, T2013/9, August 2013, TRAIL Thesis Series, the Netherlands

Wijnen, R., Decision Support for Collaborative Airport Planning, T2013/8, April 2013, TRAIL Thesis Series, the Netherlands

Wageningen-Kessels, F.L.M. van, Multi-Class Continuum Traffic Flow Models: Analysis and simulation methods, T2013/7, March 2013, TRAIL Thesis Series, the Netherlands

Taneja, P., The Flexible Port, T2013/6, March 2013, TRAIL Thesis Series, the Netherlands

Yuan, Y., Lagrangian Multi-Class Traffic State Estimation, T2013/5, March 2013, TRAIL Thesis Series, the Netherlands

Schreiter, Th., Vehicle-Class Specific Control of Freeway Traffic, T2013/4, March 2013, TRAIL Thesis Series, the Netherlands

Zaerpour, N., Efficient Management of Compact Storage Systems, T2013/3, February 2013, TRAIL Thesis Series, the Netherlands

Huibregtse, O.L., Robust Model-Based Optimization of Evacuation Guidance, T2013/2, February 2013, TRAIL Thesis Series, the Netherlands

Fortuijn, L.G.H., Turborotonde en turboplein: ontwerp, capaciteit en veiligheid, T2013/1, January 2013, TRAIL Thesis Series, the Netherlands

About the author

Curriculum Vitae

Yubin Wang was born in the town of Qianjiang, in the provide of Hubei, China, where he attended primary school, middle school and high school. From September 1998 till July 2002 he studied Mechanical Engineering at the Civil Aviation University of China located in Tianjin. After he graduated with a bachelor degree, I worked at China Southern Airlines as a mechanical engineer for one and half years. From August 2004 till July 2006 he studied Systems and Control Engineering at Delft University of Technology. During the master study, he received a DELTA scholarship. From November 2006 till April 2013, he worked at Trinité Automation as a software developer. Trinité Automation participated in a joint European Union project called Control for Coordination of Distributed Systems with Delft University of Technology, from May 2008 till September 2011. Since then he pursued his PhD research. During the period he authored 26 papers, including 3 journal papers, 1 book chapter and 22 conference papers. From April 2013 till now, he has been employed at Lockheed Martin CFT as an aircraft simulation software engineer.

Publications

Journal Articles

- Wang, Yubin, J. H. van Schuppen, and Jos L. M. Vrancken. Prediction of Traffic flow at the Boundary of a Motorway Network. IEEE Intelligent Transportation Systems Transactions and Magazine. September 2013.

- Wang, Yubin, J. H. van Schuppen, and Jos L. M. Vrancken. On-line distributed prediction of traffic flow in a large-scale road network. Simulation Modelling Practice and Theory, June 2014.

- Wang, Yubin, Jos L. M. Vrancken, F. Ottenhof, and M. Valé. Next generation traffic control in the Netherlands. Traffic Engineering and Control, 3(8):315–319, 2012.

Book Chapter

- Wang, Yubin, J. H. van Schuppen, and Jos L. M. Vrancken. Prediction of Traffic Flow in a Road Network. Coordination Control of Distributed Systems. Lecture Notes in Control and Information Sciences Volume 456, 2015, pp 45-52.

Peer-reviewed Conference Contributions

- dos Santos Soares, Michel, Jos L. M. Vrancken, Yubin Wang. Software Product Line Architecture for Distributed Real-Time Systems. 2009 International Conference on Software Engineering Theory and Practice, September 2009.

- Wang, Yubin, Jos L. M. Vrancken, and M. S. Soares. Road network representation using Dijkstra's shortest path algorithm. ITS conference in Sweden 2009, Nov. 2009.

- dos Santos Soares, Michel, Jos L. M. Vrancken, Yubin Wang. A Common Architecture to Develop Road Traffic Management Systems. ITS conference in Sweden 2009, Nov. 2009

- dos Santos Soares, Michel, Jos L. M. Vrancken, Yubin Wang. Application of publish-subscribe middleware to road traffic measurements visualization. ICNSC, pages 329–333, March 2009.

- Wang, Yubin, J. Vrancken, and M. Soares. Intelligent network traffic control by integrating top-down and bottom-up control. In Control and Decision Conference, 2009. CCDC 2009 Chinese, pages 1557–1562, June 2009.

- Wang, Yubin, Jos L. M. Vrancken, and M. S. Soares. Road network representation software generator. In 12th IFAC Symposium on Control in Transportation Systems, pages 190–195, September 2009.

- Wang, Yubin, Yuan Yuan, and Jos L. M. Vrancken. Traffic parameter estimation on motorway networks by combination of filtering techniques. In Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics, pages 3658–3662, October 2009.

- Wang, Yubin, Jos L. M. Vrancken, and M. S. Soares. A distributed simulator for road network control. In Proceedings of IEEE International Conference on Networking, Sensing and Control, pages 20–25, Aril 2010.

- dos Santos Soares, Michel, Jos L. M. Vrancken, Yubin Wang. Architecture-based development of road traffic management systems. ICNSC, pages 26–31, April 2010.

- Wang, Yubin, J. Vrancken, M. Vale and, and M. Davarynejad. Integration of urban and freeway network control by using a scenario coordination module. In Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, pages 671–676, September 2010.

- Wang, Yubin, Jos L. M. Vrancken, and Mohsen Davarynejad. Implementing scenarios coordination for road network traffic control. In Proceedings of the 2010 IEEE International Conference on Systems, Man and Cybernetics, pages 230–234, October 2010.

- Wang, Yubin, J. H. van Schuppen, and Jos L. M. Vrancken. Adaptive prediction of traffic flow into a motorway network. In Proceedings of the ITS World Congress, October 2011.

- Wang, Yubin, Jos L. M. Vrancken, and Paul Seidel. Measure travel time by using bluetooth detectors on freeway. In Proceedings of the ITS World Congress, October 2011.

- Wang, Yubin, Jos L. M. Vrancken, and Paul Seidel. Tracing travel time by using Bluetooth data on freeway. In Proceedings of the ITS Europe Congress, 2011.

- Davarynejad Mohsen, Yubin Wang, Jos L. M. Vrancken and Jan van den Berg. Multi-phase time series models for motorway flow forecasting. In Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on, pages 2033–2038, October 2011.

- Wang, Yubin, Jos L. M. Vrancken and Frank Ottenhof. Integrating Bottom-up Traffic Control into the Scenario Coordination Module. In Proceedings of the ITS Europe Congress, 2011.

- Wang, Yubin, Jos L. M. Vrancken, Frank Ottenhof and Fangfang Zheng. Comparison of ITS business in China and in The Netherlands. In Proceedings of the 2011 COSA Conference, 2011.

- Wang, Yubin, Jos L. M. Vrancken, and Ying Ma. Evaluation of the scenario coordination module, a traffic network management control system. In Proceedings of the ITS World Congress, October 2012.

- Wang, Yubin, J. H. van Schuppen, and Jos L. M. Vrancken. On-line distributed prediction of traffic flow in a large-scale traffic network. In Proceedings of the ITS World Congress, October 2012.

- Wang, Yubin, J. H. van Schuppen and Jos L. M. Vrancken. Parallelized Traffic Prediction with a Nonlinear Model. In Proceedings of the ITS World Congress, October 2013.

- Vrancken, J.L.M., Yubin Wang and J.H. van Schuppen. QHM: The Quantitative Hierarchical Model, A scalable framework for road traffic network management. In Proceedings of the ITS World Congress, October 2013.
- Vrancken, J.L.M., Yubin Wang and J.H. van Schuppen. QHM: The Quantitative Hierarchical Model for Network-level Traffic Management. In Intelligent Transportation Systems (ITSC), 2013 16th International IEEE Conference on, pages 223–230, September 2013.

Participation in international project

- Boeing 767 Full Flight Simulator
  - Objective: FedEx ordered a Boeing 767 for their cargo transportation. To effectively reducing total training cost, B767 full flight simulator will be designed and produced in order to support the different stages of a flight training program.
  - Responsibility: Scrum master, software designer and developer
  - Website: www.sim-industries.com
  - Period: April 2013 - Now

Participation in European projects

- Reducing Environmental Footprint based on Multi-Modal Fleet management Systems for Eco-Routing and Driver Behaviour Adaptation (REDUCTION)
  - Objective: REDUCTION aims at combining vehicular and ICT technologies for collecting and analyzing historic and real-time data about driving behaviour, routing information, and the associated carbon emissions measurements.
  - Responsibility: Researcher and chief designer of a green-routes system
  - Website: www.reduction-project.eu
  - Period: Sep. 2011 - Aug. 2014
- Control for Coordination of Distributed Systems (C4C)
  - Objective: C4C aims at developing control for coordination of distributed systems for five case studies with respect to control theory, communication networks, and computation.
  - Responsibility: Researcher, chief designer of a large-scale traffic prediction model and C++ developer.
  - Website: www.c4c-project.eu
  - Period: May 2008 - July 2011

Participation in Dutch National projects

- TrafficLink 2.0
  - Objective: This project aims at developing a new generation traffic information and control system with hierarchical layers.
  - Responsibility: Chief designer of the system and C++ developer.
  - Period: Sep. 2011 - May 2012
- Scenario Coordination Module (SCM)
  - Objective: SCM aims at developing a traffic control system to coordinate traffic flow between Amsterdam urban roads and motorway.
  - Responsibility: Chief designer of a large-scale traffic control system with 900 different scenario's and C++ developer.
  - Period: Mar. 2009 - Mar. 2010
- PIMVLO
  - Objective: PIMVLO aims at developing a system to monitor traffic states of Noord-Holland.
  - Responsibility: C++ developer.
  - Period: Jan. 2007 - Sep. 2008

Implemented software

- Road network representation software generator.
- Automatical incident detection algorithm on motorways.
- Average travel time calculation by using moving average algorithm based on data from Bluetooth detectors.
- Scenario coordination module.
- Traffic state estimation by using Treiber filter.
- Split factor estimation by using Kalman filter.
- Traffic prediction model in a distributed environment.
- New generation traffic management software: TrafficLink 2.0.
- Distributed prediction in AgentScape.