

**Efficient model predictive control for  
large-scale urban traffic networks**

S. Lin



# **Efficient model predictive control for large-scale urban traffic networks**

## **Proefschrift**

ter verkrijging van de graad van doctor  
aan de Technische Universiteit Delft,  
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,  
voorzitter van het College voor Promoties,  
in het openbaar te verdedigen op dinsdag 19 april 2011 om 12.30 uur  
door Shu LIN,  
Master of Science in Control Engineering, Shandong University,  
geboren te Chengde, Hebei, China.

Dit proefschrift is goedgekeurd door de promotoren:

Prof.dr.ir. B. De Schutter

Prof.dr.ir. J. Hellendoorn

Prof.dr. Y. Xi

Samenstelling promotiecommissie:

Rector Magnificus

Prof.dr.ir. B. De Schutter

Prof.dr.ir. J. Hellendoorn

Prof.dr. Y. Xi

Prof.dr. N. Geroliminis

Prof.dr.ir. C. Tampère

Prof.dr.ir. S.P. Hoogendoorn

Prof.dr.ir. B. Immers

voorzitter

Technische Universiteit Delft, promotor

Technische Universiteit Delft, promotor

Shanghai Jiao Tong Universiteit, promotor

École Polytechnique Fédérale de Lausanne

Katholieke Universiteit Leuven

Technische Universiteit Delft

Technische Universiteit Delft

The research described in this thesis was supported by a Chinese Scholarship Council (CSC) grant, Delft Center for Systems and Control, and the Transport Research Centre Delft. Shanghai Jiao Tong University made important contributions to the work described in this dissertation.

**TRAIL Thesis Series T2011/3, the Netherlands TRAIL Research School**

P.O. Box 5017

2600 GA Delft, The Netherlands

T: +31 (0) 15 278 6046

T: +31 (0) 15 278 4333

E: info@rstrail.nl

Published and distributed by: S. Lin

E-mail: lisashulin@gmail.com

ISBN 978-90-5584-136-3

Keywords: urban traffic network control, coordinated traffic control, urban traffic network modelling, model predictive control

Copyright © 2011 by S. Lin

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 of the author.

Printed in the Netherlands

# Acknowledgements

In the last four years, during the time when I was working towards this thesis, there were both cheerful moments and tough moments. All these moments made this part of my life more colorful and more meaningful. I owe a lot of thanks to the people who guided me, who help me, who accompanied me, who shared the cheerful and tough moments with me through these four years.

First, I would like to thank my supervisors for providing me this great opportunity to pursue all this, and for having been always supporting me in these years. It was my great pleasure to have Prof. De Schutter as my promoter. I have really appreciated the support, the inspiration, and the trust he gave me. It was always very efficient, fruitful, and pleasing when working with him. I am also very grateful to Prof. Hellendoorn for his belief on me that I could achieve this, which was always the great encouragement for myself in the last two years. I also want to thank Prof. Xi for giving me this research task, and leading me through the confuses. I have sincerely appreciated the freedom, the trust, and the unconditional support I have been given by him.

Next, I wish to thank the people who offered me help related to this thesis. I would like to thank Monique, whose work guided me stepping into the research field of this thesis, and Andreas, for being interested in my work and for giving me lots of valuable suggestions, and Solomon, for our pleasant cooperation. Thanks to Yuping for daily discussions. Thanks to Will for helping me solve difficult programming problems. A thank you to Maarten. I would also like to thank all my committee members, who spent their time and efforts on reading the thesis and giving their valuable feedbacks.

I would also like to thank the people who accompanied me through all the four years. Thanks to Lakshmi, who shared happy and sad moments with me, and made the life in Delft less boring. Many thanks to my friends in China, Yan Zuo, Lin Liu, Yan Zhang, Lihui Cen, Dewei Li, Xiaoli Li, Pengyuan Zheng, He Huang, Jianbo Lu, Zhao Zhou, and Haibin Shao. Many thanks to my friends in Delft, Aydin, Yelmer, Hong, Zhe, Xiukun, Jianfei, Felipe, Hans, Alfredo, Dang, Almo, Zulkifli, Alina, Samira, Katerina, Noortje, Rudy, and all my dear colleagues in DCSC. Special thanks to Kitty, Ellen, Esther, and Saskia.

Finally, I wish to thank my family for always standing on my side. I would like to thank my parents for bringing me to the world, and always supporting me to pursue my dreams. Especially my mother, thank her for giving me unconditional love, which is always a stream of warm energy for me, no matter where I am. I would also like to thank my husband for always understanding, supporting, listening, and sacrificing to give me the freedom that I need.

Shu Lin, Delft, April 2011.



# Contents

<b>Acknowledgements</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview of research background . . . . .	1
1.2 Motivation . . . . .	4
1.3 Scope and contributions of the thesis . . . . .	5
1.4 Structure of the thesis . . . . .	6
<b>2 Coordinated Traffic Control — The State of the Art</b>	<b>9</b>
2.1 Introduction . . . . .	9
2.1.1 Scope and aims . . . . .	9
2.1.2 Overview . . . . .	10
2.2 Network coordination approaches . . . . .	10
2.2.1 Optimal control . . . . .	10
2.2.2 MPC . . . . .	12
2.2.3 Rule-based strategies . . . . .	19
2.2.4 Case-based strategies . . . . .	26
2.2.5 Anticipatory control strategies integrated with traffic assignment . .	27
2.2.6 Alternative approaches . . . . .	29
2.3 String coordination approaches . . . . .	31
2.4 Comparison . . . . .	33
2.5 Summary . . . . .	34
<b>3 Framework for MPC Control of Large-scale Urban Traffic Networks</b>	<b>37</b>
3.1 Network-wide control structures . . . . .	38
3.1.1 Centralized control structure . . . . .	38
3.1.2 Decentralized control structure . . . . .	39
3.1.3 Distributed control structure . . . . .	40
3.1.4 Hierarchical control structure . . . . .	41
3.2 General problem formulation . . . . .	43
3.3 Two coordination algorithms . . . . .	47
3.3.1 Interaction balance principle . . . . .	48
3.3.2 Interaction prediction principle . . . . .	50
3.4 Summary . . . . .	52

<b>4</b>	<b>Macroscopic Spatiotemporal Discrete Urban Traffic model</b>	<b>55</b>
4.1	Introduction . . . . .	55
4.2	Notations . . . . .	57
4.3	Discrete-time delay . . . . .	59
4.4	Spatiotemporally discrete urban traffic model . . . . .	61
4.4.1	Traffic dynamics on a link . . . . .	61
4.4.2	Synchronization between two intersections . . . . .	63
4.5	CFL condition for urban traffic models . . . . .	64
4.6	BLX model and S model . . . . .	67
4.6.1	BLX model . . . . .	67
4.6.2	Simplified model (S model) . . . . .	68
4.7	Model assessment . . . . .	70
4.8	Summary . . . . .	74
<b>5</b>	<b>Subnetwork MPC Controllers</b>	<b>77</b>
5.1	Introduction . . . . .	77
5.2	Model Predictive Control: General framework . . . . .	78
5.3	MPC for traffic subnetworks . . . . .	80
5.3.1	Prediction model . . . . .	82
5.3.2	Optimization problem . . . . .	82
5.3.3	Rolling horizon . . . . .	83
5.4	Case studies . . . . .	83
5.4.1	Model test . . . . .	84
5.4.2	Urban subnetwork control using MPC . . . . .	86
5.5	Summary . . . . .	93
<b>6</b>	<b>Fast MPC for Urban Traffic Subnetworks via MILP</b>	<b>95</b>
6.1	Introduction . . . . .	95
6.2	Rules for equivalent transformation into MLD model . . . . .	96
6.2.1	Preliminaries . . . . .	96
6.2.2	Equivalent transformation into MLD model . . . . .	97
6.3	Reformulation of the urban traffic model . . . . .	98
6.3.1	Model reformulation into mixed-integer linear model . . . . .	98
6.3.2	Reformulation of the model synchronization . . . . .	100
6.3.3	Link time delay assumption . . . . .	102
6.4	MILP-based MPC controller . . . . .	103
6.5	S* model-based MPC controller via MILP . . . . .	104
6.5.1	S* model . . . . .	104
6.5.2	S* model-based MPC controller . . . . .	104
6.6	Case study . . . . .	104
6.7	Conclusion . . . . .	111
<b>7</b>	<b>Integrated MPC for the Reduction of Travel Delays and Emissions</b>	<b>113</b>
7.1	Introduction . . . . .	113
7.2	Microscopic traffic emission and fuel consumption model . . . . .	114
7.3	Integrated traffic flow, traffic emission, and fuel consumption model . . . . .	116



---

7.3.1	Urban traffic behaviors for individual vehicles . . . . .	116
7.3.2	Integrated VT-S traffic emission and fuel consumption model . . . . .	118
7.4	Objective function . . . . .	123
7.5	Case study . . . . .	124
7.6	Conclusions . . . . .	127
<b>8</b>	<b>Conclusions and Recommendations</b>	<b>131</b>
8.1	Summary of conclusions . . . . .	131
8.2	Recommendations for future research . . . . .	134
8.2.1	Recommendations for the thesis . . . . .	134
8.2.2	Recommendations for the field . . . . .	136
<b>A</b>	<b>Details for the MILP-based MPC Controller</b>	<b>139</b>
	<b>Bibliography</b>	<b>141</b>
	<b>TRAIL Thesis Series Publications</b>	<b>151</b>
	<b>Samenvatting</b>	<b>153</b>
	<b>Summary</b>	<b>157</b>
	<b>Curriculum Vitae</b>	<b>161</b>



# Chapter 1

## Introduction

Since the emergence of modern transportation, human life has become much more efficient and convenient than before. More and more people have their own private vehicles, and have more freedom to travel to any destination they want at any time they like. However, if the number of vehicles intending to travel on roads grows larger and larger, but the capacity of these roads is limited, or could not increase as fast as the growing of the number of the vehicles, then the drivers on roads cannot drive as free as before anymore. Therefore, in order to keep the public roads used in a well-organized way by all the drivers, it is very important to adopt traffic control systems to manage transportation in a good manner.

### 1.1 Overview of research background

Since 1914, when the first traffic control signal was implemented, traffic control has been developed in several aspects. From the control point of view, it developed from fixed-time controllers to traffic-responsive controllers, from model-free controllers to model-based controllers, and from heuristic controllers to optimization-based controllers. The controllers developed so that they can not only deal with under-saturated traffic, but can also work with saturated or even over-saturated traffic. The controlled area developed from initially isolated intersections to a string (a long road with multiple intersections on it), and then to a whole traffic network.

At the beginning of the development of traffic control, fixed-time control were used at intersections. In each intersection, the length of the green time durations is then always fixed, or at least fixed during time segments of the whole day. The fixed-time control actions are predefined according to the historical traffic information. However, fixed-time control is a kind of open-loop controller, which cannot adapt its control actions to the current traffic condition. Therefore, traffic-responsive control has been introduced, along with the application of a variety of detectors (loop detectors, video, etc.) to measure the traffic information. Traffic-responsive control belongs to the feedback control category, which can adjust the control actions according to the currently measured traffic conditions.

Either for the fixed-time control or the traffic-responsive control, the control strategies are not built on traffic models, but on the historical traffic information or the currently measured traffic information. These control strategies can only consider the past and the current

traffic condition, but cannot look ahead into the future. In order to avoid this, traffic models are applied in traffic control to predict the future traffic states, which results in model-based traffic control strategies. At the beginning, the traffic models used were derived inductively through traffic data identification. After that, more elaborate traffic models, which are deductively derived to describe the physical dynamics of traffic flows, were applied, and they provided more accurate predictions. However, generally speaking, the dynamic traffic models are also more computationally complex, thus need more computing time. Therefore, it is very important to find a trade-off between the accuracy and the computational complexity of the model, so that the model-based controllers can make better control decisions and also keep being applicable in real-life traffic.

Before using traffic models, the traffic control decisions were mainly made heuristically, e.g. in traffic-responsive control, if the measured length of the vehicle queue is getting longer, then more green time will be allocated to the corresponding traffic stream. But, as soon as traffic models are used, it is not necessary to make traffic control decisions heuristically anymore. Optimization tools can be made to search for the best sequence of future control decisions, based on the traffic information predicted by the models. These kind of control strategies are model-based optimization control methods, which can predict far into the future and make the best current control decisions from a long-term point of view, and then roll the prediction horizon forward and repeat solving the control problem again. Model Predictive Control, which is selected as the control method of this thesis, belongs to this category.

Initially, traffic intersections were controlled separately by local controllers in a decentralized structure [104, 122, 127]. For the local controllers, only local traffic information is taken into consideration, and no interactions among each other. So, when the traffic is in an under-saturated scenario with a low density, local traffic controllers are enough to easily regulate the local traffic and maintain it in an organized situation. However, if the traffic density grows higher or the traffic demands are provided unevenly from different directions, then it is not enough to just make decisions based on the local information. Because better local control performance does not always mean better global control performance, i.e. sometimes local traffic delay can be reduced, but at the cost of even more traffic delay and congestion emerged somewhere else in the same traffic network. Therefore, it is necessary to investigate traffic control strategies to coordinate road strings (e.g. a highway or a main road in urban areas), and even to coordinate the traffic in an area of a network.

A number of coordinated urban network control strategies have already been developed [56, 104, 112]. Fixed-time coordinated control strategies make control decisions off-line based on the traffic flow data collected and stored in the past. Traffic-responsive coordinated control strategies can in real time measure the traffic states in the network, and adapt the control schemes according to the current measured traffic states. Model-based coordinated control strategies [3, 17, 42, 44, 49, 60, 104, 115, 121] do not only introduce in feedback control so as to adjust in real time the control decision according to the current detected traffic states, but also predict into the future using prediction model to make decisions good also in a long term run. The structures for the coordinated control strategies can be centralized, distributed, or hierarchical. Centralized coordination control strategy optimizes the whole traffic network and searches for a global optimal solution for the network. Distributed coordination control strategy allocates the control efforts to each local traffic controller, and coordinate the local controllers through information exchange. Hierarchical

coordination control strategy divides the overall complex control problem for large-scale system into multiple levels, on each level, a specific control problem will be solved.

There are already well-known coordinated traffic-responsive control strategies for urban traffic networks. SCOOT [21, 112] and SCATS [91] are widely used in many big cities around the world [104], e.g. SCOOT is used in Beijing, SCATS used in Shanghai. They are both dynamic traffic control strategies based on measured current traffic states in distributed, multi-level, hierarchical system structures. It has already been shown that these two systems work effectively in real traffic world. But, these two systems are more focusing on dynamic intersection controllers, and local coordinations that consider only a few neighbor intersections. In the 1980s and 1990s, a number of model-based optimization control strategies based on simple traffic models emerged, e.g. OPAC [49], PRODYN [44], CRONOS [17], RHODES [115], and MOTION [12], which can forecast the future traffic behavior of the network based on models. With these forecasting models, the control strategies are able to make control decisions to guarantee better performance within an area of the traffic network in a near future. A real test was realized for OPAC in Reston, USA (16 intersections) [48], and for MOTION in the center of Köln-Deutz, Germany (12 intersections) [13]. However, the models used in these control approaches are mainly simple traffic models based on the traffic data measured by upstream detectors, which to some extent limits the performance for the future. Coordinated traffic-responsive control strategies that are able to avoid parts of the on-line computational complexity, were also proposed. UTOPIA/SPOT [93] is a hierarchical system with simple local intersection controllers and a central controller for an area of urban networks. The central controller optimizes the control actions for the whole area based on the model of the network. The local controller makes the decision only based on local information, but with a penalty term to guarantee that the local decision is not too far from the central decision. Therefore, UTOPIA/SPOT avoids part of the on-line computational burden, but results in suboptimal solutions. TUC [3, 40] was proposed for controlling an urban traffic network based on the well-known simple store-and-forward model. TUC designs a feedback regulator off-line based on the store-and-forward model, and on-line derives the traffic signals using a feedback control law by feeding it with the real-time measured traffic states. Therefore, the TUC strategy reduces the on-line computational complexity significantly by moving the time-consuming optimization off-line. Compared with the fixed-time controller, TUC can reduce the Total Time Spent by 20-54% for different scenarios [40]. TUC has been implemented in three cities — Chania, Greece (23 intersections), Southampton, UK (53 intersection), and Munich, Germany (25 intersections) — and has been proved to have good control effects [68]. However, when the real traffic conditions change, the feedback control law needs to be redesigned according to the new current traffic conditions, which is also computational complex if it occurs too frequently.

In summary, traffic control methodologies have been developed for a long period of time, and the results are fruitful. A number of urban traffic control systems have been presented in literatures or even applied in practice. Some of these systems, that were implemented in real-life traffic field, have been proved effective in practice. However, the efficiency of coordination algorithms, especially of network-wide coordination algorithms for large-scale urban traffic networks, is still needed to be further improved. In this thesis, we are going to mainly focus on the solutions to address this issue.

## 1.2 Motivation

One promising control methodology that can meet all the needs for controlling and coordinating a large-scale traffic network with a number of traffic control measures (e.g. traffic signals, ramp metering, speed limits, etc.), is Model Predictive Control (MPC) [23, 92, 108]. In the late 1970s, MPC was first proposed and successfully applied in the process industry. MPC executes model-based optimization control on-line in a rolling horizon way, and thus it has shown to be able to respond in an effective way to the disturbances in many practical process control applications. Therefore, MPC is widely accepted in the process industry.

MPC is a methodology that implements and repeatedly applies optimal control in a rolling horizon way. In each control step, an optimal control problem is solved over a prediction horizon, but only the first control sample of the optimal control sequence is implemented. Next, the prediction horizon is shifted one sample and the optimization is restarted again with new information of the measurements. The optimization is redone based on the prediction model of the process and an estimate of the disturbances.

MPC has a number of advantages for controlling large-scale traffic networks. MPC can easily deal with multi-input and multi-output problems with constraints. Therefore, different traffic control measures (e.g. traffic lights, ramp metering, speed limits, etc.) are able to be controlled and coordinated at the same time. Since MPC approach is a model-based optimization control strategy, it can combine multiple objectives into one control problem, if the optimization problem of the MPC controller is a multi-objective optimization problem. Thus, MPC can combine multiple control objectives into one control problem, so it is easy to integrate different control objectives (e.g. traffic congestion control, traffic emissions control, etc.) into MPC controllers. Moreover, due to the rolling horizon procedure, MPC becomes a closed-loop controller, which can in realtime adapt the controller by the feedback information measured from the real-life traffic. Consequently, MPC has the ability to deal with the uncertainty of the traffic system, which can be caused by the unpredictable disturbances, the (slow) variation over time of the parameters, and model mismatches in the prediction model. Another advantage of MPC is that one can easily select and replace the prediction model based on the control requirements.

However, although MPC is a well-established control method, a big difficulty to implement MPC in practice is the high on-line computational burden. When using MPC, for each time step, we have to solve an optimization problem within a limited period of time. The solving speed depends on both the scale of the optimization problem, and the features of the optimization problem itself. Of course, the solving speed can be also improved by using very fast computers, but the development of the computer hardware capability cannot always catch up with the complexity expansion of practical problems. So, we will mainly discuss about the scale of the optimization problem, and the features of the optimization problem. The scale of the optimization problem increases when the scale of the traffic network controlled grows larger, and when the length of the prediction horizon gets longer. Optimization problems differ from each other by their features. Some can be solved easily and quickly, such as linear programming, quadratic programming, and convex optimization. But, some are hard and time-consuming to solve, like nonlinear non-convex optimization. Due to the nonlinearity of most traffic prediction models, the optimization problem of the MPC controller is in general a nonlinear optimization. Therefore, when the on-line optimization problem of MPC controller is time-consuming to solve, no matter because of the

large scale of the optimization problem or because of the hard-to-solve feature of the optimization problem, the MPC controller becomes real-time infeasible in practice, even though its optimization problem is solvable in theory.

Therefore, we are going to focus on the approaches to increase the real-time feasibility of the on-line optimization problems, when we apply MPC to control large-scale urban traffic networks. The approaches investigated in this thesis are:

- Reducing the computational complexity of the urban traffic control model,
- Reformulating the on-line optimization problem so that it can be solved more efficiently,
- Dividing the network into small subnetworks, and building distributed network controllers.

In this thesis, we are going to mainly investigate the first two approaches to reduce the on-line computing time of the MPC optimization problems for MPC controllers. Regarding to the third approach, we are going to present a general framework for MPC control of large-scale urban traffic networks, and analyze the different control structures for the network-wide traffic controller. Consequently, all the centralized MPC controllers presented in this thesis are able to fit into the framework, and can act as local controllers for the urban traffic subnetworks. Some of the results can be also extended and applied to freeway networks, or even mixed freeway and urban road networks.

### 1.3 Scope and contributions of the thesis

In this thesis, we mainly focus on the coordinated control for large-scale urban traffic networks. The control method we apply is MPC. The main problem we are going to deal with is the large online computational complexity when the MPC theory is applied to the real-life traffic system. Before the story starts, a state-of-the-art literature survey on coordinated traffic control is given, which summarizes former research on this topic, and which gives some general information for the students or researchers who are interested in this topic.

When the scale of traffic networks we need to manage grows larger and larger, the control problems for the large-scale traffic networks also become more and more complex. In order to control a large-scale traffic network, and get a balance between the local performances and the global performance, the controllers should operate in a well-designed structure. Therefore, in this thesis, MPC controllers are organized and coordinated in a distributed control framework for large-scale traffic networks. Under this framework, the possible network-wide traffic control structures are discussed and analyzed.

In the distributed control structure, MPC based traffic controllers are designed as local controllers for urban traffic subnetworks. A part of the research work of this thesis is to improve the feasibility of the MPC subnetwork controllers in practice, and to improve the efficiency of the higher level algorithms when coordinating all the subnetwork MPC controllers. More efficient subnetwork MPC controllers for large-scale urban traffic networks are obtained by model reduction and optimization problem reformulating. The traffic models included in this thesis are all discussed and evaluated from a control point of view. To improve the applicability of the MPC controllers for large-scale urban traffic networks, we

are mainly focusing on the trade-off between the efficiency and the complexity of the traffic models.

Since traffic pollution is also an issue for the people living in a big metropolis, it is necessary to take traffic emissions into consideration when we are controlling urban traffic. Therefore, MPC controllers that integrated both traffic delay control and traffic emission control are also discussed in the thesis.

The main contributions of this thesis are:

- A macroscopic traffic model for urban traffic networks, called BLX model <sup>1</sup>, is established. To further reduce the computational complexity, a more simplified macroscopic new model, the S model <sup>2</sup>, is proposed. The S model is very fast, but is still able to provide all the necessary information that is needed for traffic control. We also introduce another model with an adjustable sample time step that contains features of both the BLX and the S model, and that is excellently suited to tune the trade-off between accuracy and computational complexity.
- MPC controllers are built for urban traffic networks based on the BLX and the S model. The on-line computational efficiency of the MPC controllers based on the S model is improved greatly compared to MPC based on the BLX model, and only a limited loss of control performance is incurred.
- The nonlinear optimization problem of the urban traffic MPC controllers is reformulated for the S model into a mixed-integer linear programming (MILP) problem, which can be solved very efficiently by existing MILP solvers. We also introduce a further simplification of the S model, called the S\* model, that results in a smaller MILP problem. For both the S model and the S\* model the on-line computational efficiency of the MPC controllers is further improved significantly compared to the nonlinear optimization approach.
- In order to control traffic delay as well as traffic emissions and fuel consumption in big cities, an integrated urban traffic, emission, and fuel consumption model is proposed. MPC controllers are built based on this model, which results in a balanced trade-off between minimizing travel time and reducing both emissions and fuel consumption.

## 1.4 Structure of the thesis

This thesis contains eight chapters, and this introduction makes the first chapter of the thesis. Chapter 2 is a literature survey on this research topic. Chapter 3 gives an overview of the general control framework. Chapter 4 discusses about urban traffic network models. In Chapter 5, model predictive controllers are established based on different traffic models, while Chapter 6 further reformulates the optimization problem of these MPC controllers. Chapter 7 also focuses on traffic emission problem. The conclusions can be found in Chapter 8.

---

<sup>1</sup>The macroscopic traffic model is extended from the model of M. van den Berg et al. [119], and then is revised by S. Lin and Y. Xi, thus is called BLX model.

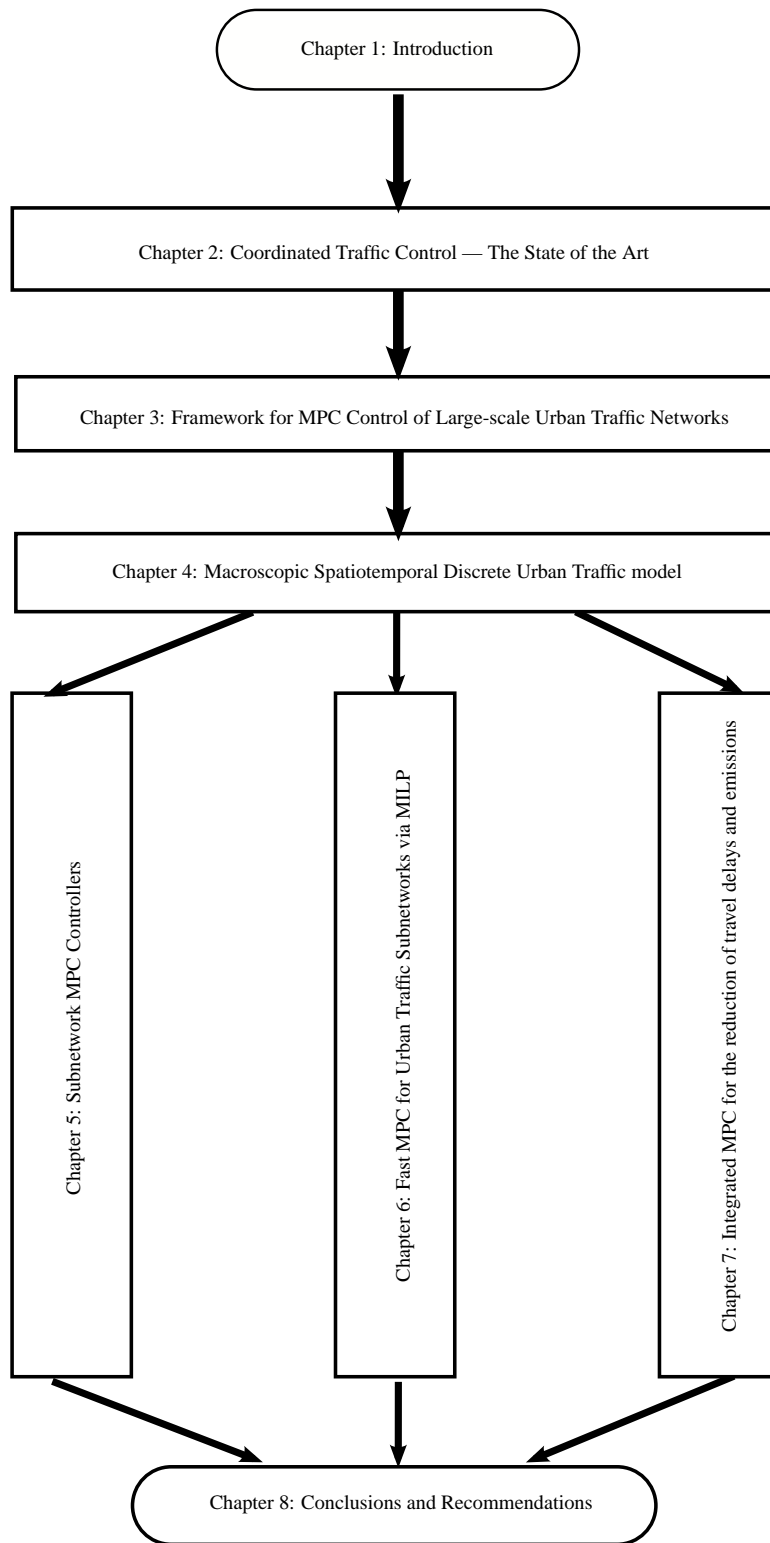
<sup>2</sup>A Simplified model proposed by S. Lin et al. [80]



The detailed content of the main chapters is summarized as follows:

- **Chapter 2** is a literature survey summarizing the state-of-the-art on coordinated traffic control, where the main existing coordinated traffic control strategies are summarized by the control methods for both freeways and surface roads. The characteristics of these coordinated traffic control strategies are discussed and compared in this chapter.
- **Chapter 3** gives the framework for constructing structured network-wide traffic controllers based on MPC. Different traffic network control structures, including decentralized structure, centralized structure, distributed structure, and hierarchical structure, are discussed in this chapter. Coordination algorithms are also presented to coordinate traffic subnetworks controlled by MPC controllers.
- **Chapter 4** discusses about the macroscopic urban traffic models, which are all spatiotemporally discrete. New models are presented which has low computational complexity, at the same time, keeps enough modeling accuracy for traffic control purposes.
- **Chapter 5** presents the method of constructing model predictive controllers for urban traffic subnetworks, based on the models discussed in Chapter 4. The MPC controllers based on different urban traffic subnetwork models are further compared in this chapter.
- **Chapter 6** presents how to reformulate the on-line optimizations of MPC controllers for urban traffic networks to increase the computational efficiency. To this aim, the former nonlinear non-convex optimization problem is reformulated into a mixed-integer linear programming optimization problem, and the reformulated optimization problem can be solved much faster than the former nonlinear non-convex optimization problem.
- **Chapter 7** focuses on integrated MPC controllers for the reduction of travel delays as well as traffic emissions and fuel consumption in urban traffic networks. The integrated functions of the MPC controller depend on the traffic model, which integrates a microscopic vehicle emission and fuel consumption model with the macroscopic traffic flow model proposed in Chapter 4.

In Fig. 1.1, an overview of all the chapters in this thesis is illustrated in a flowchart. Chapter 1 gives the motivation and a brief overview of the entire thesis. Chapter 2 summarized most of the related research work, further motivates the research of this thesis. Chapter 3 and Chapter 4 present a general framework for controlling large-scale urban traffic networks, and prediction models, which are the basis of the rest of the thesis. Based on the previous chapters, MPC subnetwork traffic controllers are designed in Chapter 5, Chapter 6, and Chapter 7, aiming at solving three problems in the urban traffic network control. Chapter 8 concludes the thesis and gives recommendations for the future research directions.



*Figure 1.1: Structure overview of the thesis*

## Chapter 2

# Coordinated Traffic Control — The State of the Art

### 2.1 Introduction

In recent years, the number of vehicles has grown larger and larger, and the requirements for traveling by vehicles are getting more and more stringent. Even though large and sound traffic networks (freeways and roads) are already constructed, traffic congestion still can not be avoided efficiently. Moreover, it is often too time and money consuming to build more common transportation infrastructures or reconstruct the ones that already exist. Therefore, traffic jams occur frequently and have a severe impact, when people need to use the common infrastructures with limited capacity at the same time, especially during rush hours. Traffic congestion can give rise to traffic delays, economic losses, traffic pollution, and so on. To reduce traffic jams and to promote efficiency in traveling, effective traffic control methods are necessary. In this context, traffic control strategies are one of the most efficient and also effective methods to solve the problem.

Since traffic control emerged, a large number of control algorithms were proposed and implemented in the field, like fuzzy control, PID control (Proportional—Integral—Derivative controller), etc. However, these algorithms are mainly focusing on controlling a single intersection or a single traffic control measure. These controllers are without global scope, and have limited control effect for the whole traffic network. As we known, traffic intersections are not isolated; the traffic states of roads in a traffic network will interact with each other; a traffic jam that happens here is maybe caused by some irregular event (e.g. an incident) that happened somewhere else in the same traffic network. Therefore, it is necessary to understand the behavior of traffic networks, and to investigate network-wide traffic coordinated control approaches that can coordinate and control traffic networks to a better performance.

#### 2.1.1 Scope and aims

This literature survey focuses on coordinated traffic control strategies, both for freeways and urban roads. By searching and summarizing the recent works, a general idea of the state-

of-the-art traffic coordination control methods is obtained. After analyzing and comparing the collected approaches, we provide insight into the characteristics of different kinds of coordinated traffic control strategies. Then, a conclusion of which approach is both effective and suitable to be implemented in the field at present is obtained, and future directions for investigation are presented.

### 2.1.2 Overview

The structure of this literature survey is as follows:

In Section 2.2 coordination strategies for traffic networks are introduced and summarized into several categories.

Although a traffic string (i.e. an artery or a freeway link) is a part of the traffic network, several papers and articles still only focus on the coordinated control problem of traffic strings. Section 2.3 therefore discusses coordinated control strategies at the string level.

In Section 2.4, the characteristics of different kinds of coordinated traffic control strategies are analyzed and compared.

In Section 2.5, the conclusions of the literature survey are provided, and possible future directions for traffic coordinated control are given.

## 2.2 Network coordination approaches

Coordinated traffic control strategies both for freeway networks and urban networks are discussed in this section. They are classified into different categories according to the control methodologies adopted, i.e. optimal control approaches, Model Predictive Control (MPC) approaches, rule-based approaches, case-based approaches, and approaches based on the network macroscopic fundamental diagram.

### 2.2.1 Optimal control

After the emergence of suitable traffic models, more advanced model-based controllers (e.g. optimal control) started to be used to coordinate freeway networks. The main idea of optimal control is to find the optimal control measures of the whole freeway network in the future by optimizing the cost function based on a network model for a certain future time horizon. The optimal control approach can coordinate the freeway network in a centralized structure. It not only can coordinate the control measures on different space locations and different time points in the future, but it can also coordinate different types of control measures (e.g. ramp metering, speed limits, and route guidance). Optimal control approaches for freeway networks and urban networks are both discussed below.

**Freeway networks** AMOC (Advanced Motorway Optimal Control) [70] and OASIS (Optimal Advanced System for Integrated Strategies) are two control software tools based on optimal control theory. They both adopt the macroscopic freeway traffic model METANET [96] as optimization model. However, because the freeway network model is nonlinear, one of the big challenges of applying optimal control is to find an efficient algorithm to solve the large-scale optimization problem. A numerical solution algorithm that is based on

a feasible-direction nonlinear optimization method, is proposed to successfully solve this problem [69, 71, 72]. The AMOC approach has been applied to the Amsterdam ring-road [70], and proved to have good coordination control effectiveness.

**Urban networks** In recent years, a number of urban traffic models have been proposed. For different urban traffic models, different optimal control approaches have subsequently been derived.

The store-and-forward model is a linear state-space model for road networks of arbitrary size, topology, and characteristics, and is given by

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \mathbf{B}\mathbf{g}(k) + T\mathbf{d}(k) \quad (2.1)$$

where  $k$  is the counter of time steps,  $T$  is the control time interval;  $\mathbf{x}(k)$  is the state vector (consisting of the number of vehicles  $x_z$  of each link  $z$ );  $\mathbf{g}(k)$  is the control vector (consisting of the green time  $g_{ji}$  of each stage  $i$  at each junction  $j$ );  $\mathbf{d}(k)$  is the disturbance vector (consisting of the demand flows  $d_z$  of each link  $z$ );  $\mathbf{B}$  is a constant matrix of appropriate dimensions containing the network characteristics (topology, saturation flows, turning rates). The linear state-space feature of the store-and-forward model opens the way to the application of a number of highly efficient optimization and control methods (such as linear programming, quadratic programming, and multivariable regulators) with polynomial complexity. Based on the store-and-forward model, an open-loop quadratic-programming control (QPC) [3] approach is developed, which can be efficiently solved by using broadly available codes of commercial software.

However, to keep the linear characteristic, the store-and-forward model is only applicable under a saturated traffic scenario. Therefore, an open-loop nonlinear optimal control (NOC) [3] approach is developed based on a nonlinear urban traffic model, that is more elaborate to describe more complex traffic dynamics. A numerical feasible-direction optimization algorithm is applied to solve the nonlinear optimization iteratively, which requires more computational complexity than QPC.

To avoid the inherent drawbacks of an open-loop structure, a linear-quadratic (LQ) optimal control approach, Traffic-responsive Urban Control (TUC) [3, 41, 68], is developed based on the store-and-forward model. Instead of optimizing the control inputs (i.e. green times), TUC optimizes the linear multivariable feedback regulator off-line, as

$$\mathbf{g}(k) = \mathbf{g}^N - \mathbf{L}\mathbf{x}(k), \quad (2.2)$$

where the feedback gain matrix  $\mathbf{L}$  results as a straightforward solution of the corresponding algebraic Riccati equation, and  $\mathbf{g}^N$  is a nominal vector for  $\mathbf{g}$ . The feedback regulator is actually a feedback control law, which is assumed to be a linear function of the traffic states  $\mathbf{x}(k)$  for the linear traffic control problem presented in TUC. The parameters of the feedback control law, i.e. the feedback gain matrix  $\mathbf{L}$ , can be obtained through off-line optimization. Then, the optimized feedback regulator can be actuated on-line to derive the new green times, fed with the real-time measured traffic states  $\mathbf{x}(k)$ , and no on-line optimization is needed.

Dynamic Intersection Signal Control Optimization (DISCO) [90] is a dynamic urban traffic optimization control approach based on the cell-transmission model. A cell-transmission

model [32, 33] is a convergent numerical approximation to the hydrodynamic model of traffic flow. It considers the entire fundamental diagram and can capture traffic phenomena such as shock waves and queue dynamics. The timing plans of the urban traffic network are derived by solving the optimization problem via a genetic algorithm. DISCO is proved to be superior to TRANSYT, especially under congested situations.

In spite of all the advantages, the optimal control approach is still open-loop. It solves the optimization problem based on the approximation of the future network disturbances, which can be inaccurate, or even be the opposite to reality when unpredictable events occur. Moreover, mismatches between the model and the real world, and inaccuracies in estimating initial traffic states can always happen. Under these circumstances, the control results derived from optimal control methods are not the best coordination control actions anymore.

### 2.2.2 MPC

Model Predictive Control (MPC) [23, 92] is a methodology that implements and repeats optimal control in a rolling horizon way. This means that, in each control step, only the first control sample 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.

Taking optimal control as the core algorithm, MPC preserves all the advantages of optimal control. It can predict and find the coordinated optimized solution for the entire network in the future. It can also coordinate different types and numbers of control measures. Due to the rolling horizon methodology, the MPC controller becomes closed-loop by adjusting the controller with a real-time feedback. The MPC controller thus obtains the ability to deal with the uncertainty of the real world, caused by unpredictable disturbances, (slow) variation over time of the parameters, and mismatch errors of the prediction model.

In principle, a centralized MPC method can maximize the throughput of the whole network, and provide network-wide coordination of the traffic control measures. However, the real-time computation complexity is a big challenge for implementing MPC controllers to traffic networks in practice. In general, the computational complexity will increase exponentially when the scale of the network grows (if the prediction model is nonlinear). To overcome this problem, different structures (e.g. decentralized and hierarchical structures) other than the original centralized structure are taken to maintain the real-time feasibility of MPC controller.

#### Centralized structure

**Freeway networks** Hegyi et al. [59, 60] apply MPC taking METANET as the predictive model to control and coordinate the freeway networks in the centralized structure (see Fig. 2.1).

To suppress shock waves, coordination of variable speed limits is studied adopting the MPC methodology. Simulations are carried out on a benchmark network consisting of a link of 12 km, where 6 segments of 1 km are controlled by speed limits. The simulation results show that the MPC controller is effective for coordinating speed limits against shock waves.

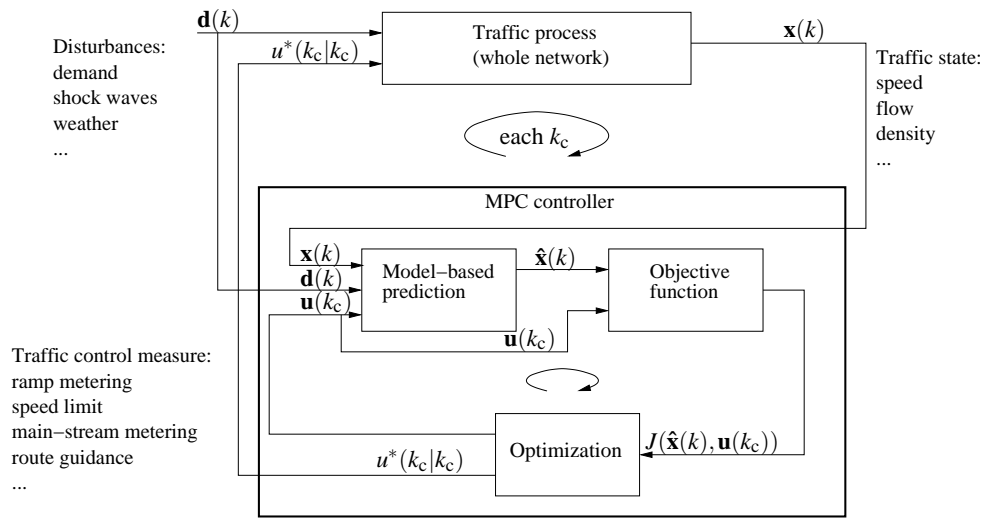


Figure 2.1: The MPC scheme for traffic control [58].

The shock wave generated from the downstream end of link is successfully eliminated by the coordinated control of the speed limits.

There are several control measures for freeways, e.g. ramp metering, dynamic speed limits, and main-stream metering. These control measures can influence each other, or may have different effectiveness in different traffic scenarios. An MPC controller is used to coordinate these different types of freeway control measures. Experiment results show that the speed limits can complement ramp metering, when the traffic demand is so high that ramp metering alone is not efficient anymore. Conclusions are also drawn that the coordinated and integrated control of speed limits and ramp metering results in a higher outflow and a significantly lower total time spent. It is also stated that the choice between speed limits and main-stream metering should be made based on the demands on the on-ramp and the freeway.

**Urban networks** In the 1980s and 1990s, a number of model-based optimization control strategies emerged: OPAC [49], PRODYN [44], CRONOS [17], and RHODES [115]. The prediction models for these strategies are similar. They mainly predict the future traffic demands at the intersections through the historical data measured from the upstream detectors or the detectors of upstream links. These strategies showed advantages compared with the traffic-responsive strategies that do not use any predictions. However, this kind of prediction models are limited in the length of the time horizon over which they can predict. The longest prediction horizon is the time taken by the vehicles running from the upstream detector to the stop-line of the intersection. Therefore, the control strategies cannot look ahead far enough due to this limitation.

In recent years, some macroscopic urban traffic models were developed for establishing more elaborate and effective model-based rolling horizon control approaches. These models can describe the traffic dynamic mechanics of the whole urban traffic network, and

overcome the drawbacks of the previous models.

The model proposed in [8] and extended in [42] is computationally intensive and it can describe different traffic scenarios, but it is also complicated and needs historical data to estimate the coming traffic flow rate of each intersection. A controller based on a rolling horizon methodology is developed by optimizing this traffic model fed with the historical data from last iteration.

The model proposed by Kashani et al. [66] has a lower modeling power, and in particular cannot depict scenarios other than saturated traffic. The model of van den Berg et al. [58, 119, 120] is an extension of the Kashani model that is capable of simulating the evolution of traffic dynamics in all traffic scenarios (unsaturated, saturated, and over-saturated traffic conditions) by updating the discrete-time model in small simulation steps. This model provides a good trade-off between accuracy and computational complexity. An MPC controller is developed based on the model [58, 120], which gives good control effects.

**Mixed freeway and urban networks** Freeway networks and urban networks are closely connected. Congestion on the freeway often causes spill back of urban queues, slowing down the urban traffic, and vice versa. As a consequence, control measures taken in one of the two areas can have a significant influence on the other area. By connecting the urban traffic model [120] and the freeway traffic model METANET with the on-ramp and off-ramp model, an integrated MPC controller is established to coordinate the mixed freeway and urban network [121]. The coordinated control approach is proved to have a high performance.

### Distributed structure

**Freeway networks** A distributed control structure can be developed to avoid the exponential growth of the computational complexity for the centralized MPC, when the network scale keeps on increasing. Game theory has been introduced to find the optimal coordination of ramp metering and variable speed limits in a large-scale freeway traffic network [54]. The large-scale freeway traffic network is then decomposed into subproblems, each of which is controlled by MPC based on the METANET model. Game theory (i.e. sample fictitious play) coordinates the sub-MPC controllers. Each player (sub-MPC controller) finds its best strategy assuming that other players play the known strategy, which can be drawn from the history of their past plays. Thus, players learn to know other player's strategies iteratively.

The sample fictitious play (SFP) algorithm is explained as [54]:

1. **Initialization** - The set of initial strategies is chosen randomly for each player and stored as historical data.
2. **Sampling** - For each player, a strategy is drawn from the history of plays arbitrarily with equal probability.
3. **Best-reply** - Each player computes its best reply or strategy assuming that other players play the strategies drawn in the previous step.
4. **Store** - The best replies obtained in Step 3 are stored in the history.



5. **Stop Condition** - It is checked whether the stopping criterion is satisfied. If so, then stop, else go to Step 2.

In [54], for a case study of 4 players (i.e. 2 on-ramp metering controllers and 2 speed limit controllers), the SFP-MPC, which can compute in parallel, reduces the optimization time by 81.1%, from 106 s to 20 s, compared with the original centralized MPC controller.

**Urban networks** Game theory is also used as distributed control method for urban networks in CoSIGN [25]. The problem of finding optimal coordinated signal timing plans for a large number of traffic signals is a challenging problem because of the exponential growth in the number of joint timing plans that need to be explored as the network size grows. However, if we decompose the problem into smaller subproblems, we may be able to find a sufficiently good solution in a reasonable amount of time. The decomposition of the problem can be accomplished by assuming that each signal in each period is an independent decision maker. The effect is to reduce an exponential number of alternatives to consider to a linear number by solving small subproblems, and coordinating them in an iterative way. To coordinate the decision makers (traffic signals), game theory (fictitious play [22, 76, 94, 98]) is applied in [25]. If each decision maker who controls a time period for a signal is viewed as a player in the game, and the average travel time of all vehicles in the traffic network is viewed as a common payoff for every player, the coordinated-traffic-signal-control problem can then be represented as a game of identical interests. A joint decision is called a Nash equilibrium if no individual player can improve its payoff by unilaterally deviating from the original joint decision. The Nash equilibrium can be viewed as a coordinated local optimum. The equilibrium situation is not always uniquely determined and it is even possible that oscillations occur. The equilibrium situation that is achieved after an iterative adjustment of traffic control is not always a system optimum [118]. An event-based mesoscopic deterministic traffic simulator, INTEGRATION-UM, is used as the traffic simulating model for the coordination algorithm.

### Hierarchical structure

**Freeway networks** Due to the open-loop nature of the optimal control approach AMOC, the derived optimal control actions are deteriorated by all kinds of system errors, such as initial states estimation error, future disturbance prediction error, model parameter mismatch error, and unpredictable incident errors. Therefore, Kotsialos et al. [73] proposed an MPC approach based on the AMOC algorithm under a hierarchical control structure to avoid the drawbacks by introducing in the rolling horizon procedure.

The hierarchical control structure consists of three basic layers (see Fig. 2.2): the Estimation/Prediction Layer, the Optimization Layer, and the Direct Control Layer. The Estimation/Prediction Layer receives historical information and real-time detected traffic states to generate the current state estimation and future predictions of the disturbances for the next layer. The Optimization Layer (AMOC) optimizes the control state trajectory over a future time horizon based on the initial states estimation and future disturbance prediction from the upper layer. Then, in the Local Direct Layer, the local ALINEA (Asservissement LINéaire d'Entrée Autoroutière) controller is adapted by the real-time optimized traffic set-points obtained from the upper Optimization Layer. ALINEA is a local proportional ramp metering control strategy with feedback [103].

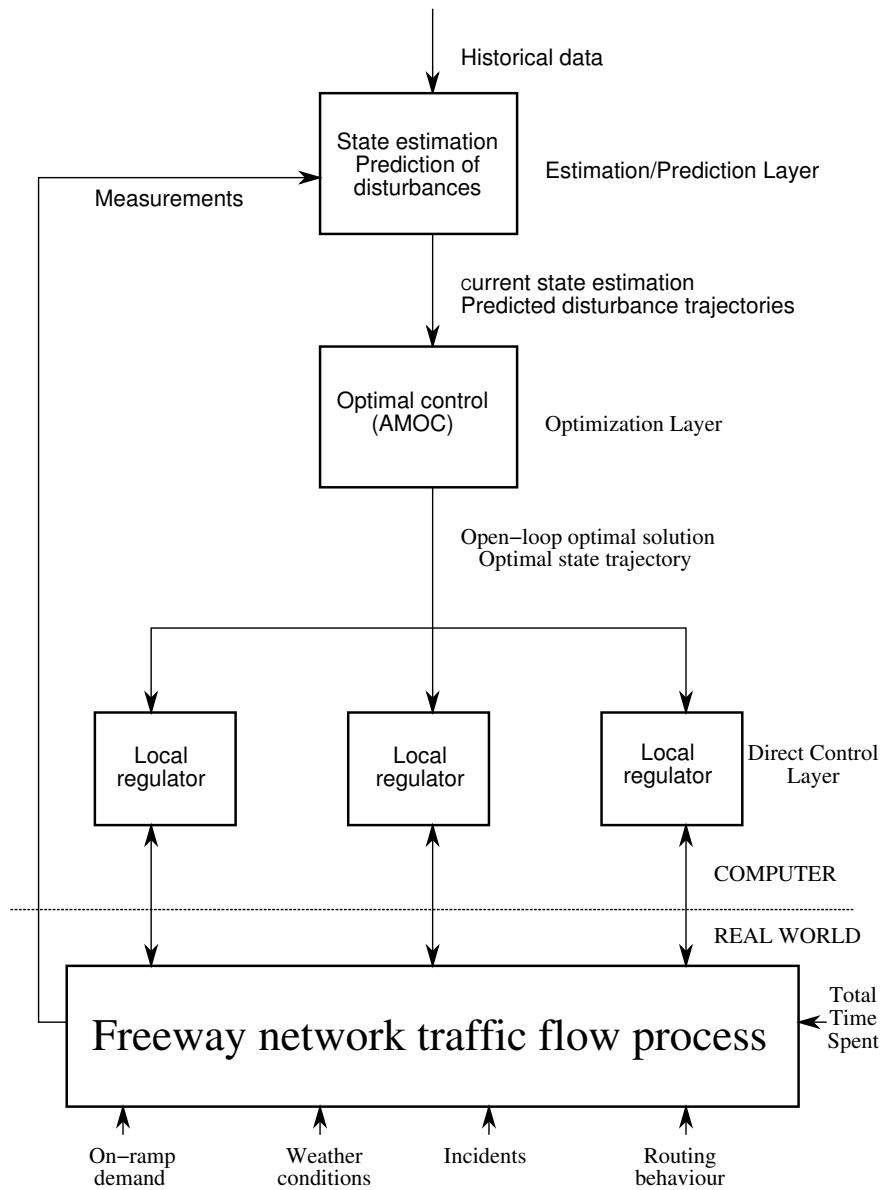


Figure 2.2: Hierarchical structure [73]

The rolling horizon hierarchical coordinated control has been applied to the Amsterdam ring-road, and outperforms the local ramp metering approach in terms of both efficiency and equity [73]. The Amsterdam test is a real test. The combination of AMOC with ALINEA preserves the positive features of both and cancels their deficiencies.

**Urban networks** For centralized control strategies, the real-time computational complexity will increase exponentially when the scale of the urban traffic network grows, which makes these strategies real-time infeasible for larger networks. Distributed control strategies can avoid this problem, but more effort needs to be taken to coordinate the local controllers. Therefore, hierarchical control structures are adopted to address the problem by implementing the centralized control algorithms as local controllers for an intersection or a small subnetwork [48, 97, 122, 123]. A hierarchical control structure divides the complex control problem of a large traffic system into different control levels or layers. In different layers, control problems with different focuses are solved. Moreover, control problems with different details are addressed in different levels, e.g. the lower control level mainly focuses on local control in a more elaborate way, and the higher control level deals with network-wide coordinated control in a more general way.

1. Virtual-Fixed-Cycle OPAC (VFC-OPAC) [48] is the hierarchical version of OPAC, which consists of a three-layer control architecture as shown in Fig. 2.3. The OPAC (Optimized Policies for Adaptive Control) control strategy is the adaptive algorithm implemented as the local controller of the hierarchical framework. The Local Control Layer implements the OPAC rolling horizon procedure: it continuously calculates optimal switching sequences for the predictive horizon, subject to the VFC constraint communicated from the Upper Synchronization Layer. The Coordination Layer, optimizes the offsets at each intersection (once per cycle). The Synchronization Layer, calculates the network-wide virtual-fixed-cycle (once every few minutes as specified by the user). The cycle length can be calculated separately for groups of intersections, as desired. Over time the flexible cycle length and offsets are updated as the system adapts to changing traffic conditions.

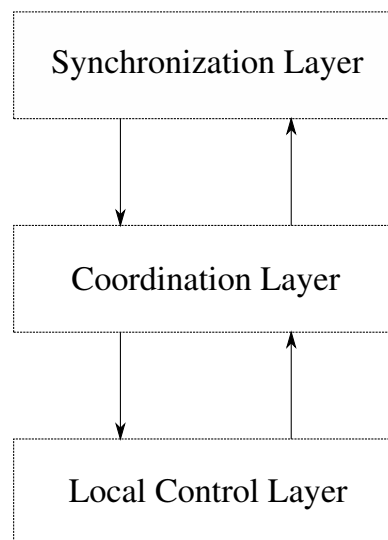


Figure 2.3: Control structure in VFC-OPAC (adopted from [48])

2. The RHODES system is developed into a multi-level hierarchical structure [97], see

Fig. 2.4. There are three levels: At the highest level, there is a dynamic network loading model that captures the slow-varying characteristics of traffic, which are caused by the network geometry, e.g. road closures and construction. At the middle level, network flow control, which making decisions according to the prediction and estimation of the traffic flow loads on the roads, is actuated to coordinate road network. At the bottom level, intersection control is carried out by applying a model-based rolling horizon optimization approach.

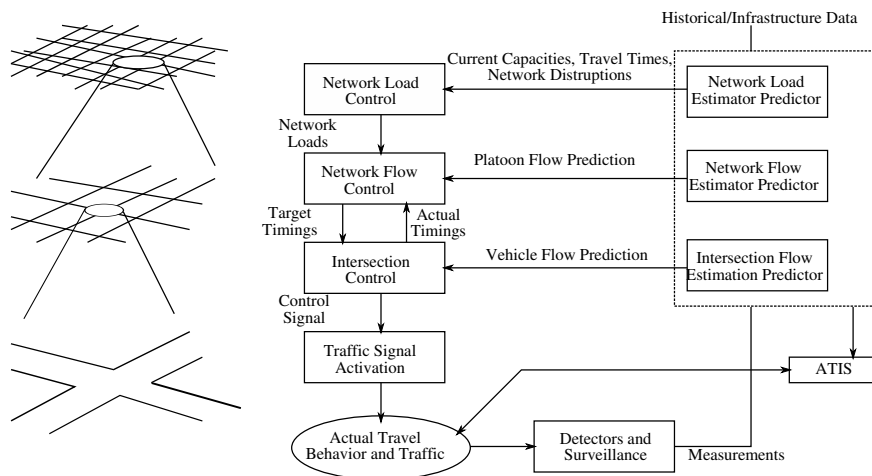


Figure 2.4: The RHODES hierarchical architecture (adopted from [97])

At the middle level, a string coordination approach, named REALBAND, is adopted to coordinate intersections. Dynamics of platoons, which are characterized in terms of size (number of vehicles) and speed, are predicted. When two (or more) platoons are predicted to arrive at the same intersection and request opposing signal phases, conflicts happen. Based on the predicted platoon movement in a pre-defined time horizon, REALBAND then searches for the best solution to overcome the conflicts using a decision tree. This decision tree lists all the possible resolutions for the conflicts, and then makes a decision. Thereafter, the REALBAND decisions, i.e. phase durations, are used as constraints to the optimization of the intersection control logic. The intersection controller will decide the phase start and end times based on recent, and more accurate observations of the vehicles in each platoon.

3. In [122] a hierarchical traffic control structure is developed. For the bottom level, a multi-agent approach is applied to reduce the computational complexity, and to add scalability to the control system. For the upper level, the local controllers are coordinated in both the microscopic and the macroscopic way. The traffic control problem is divided into several loosely coupled subproblems, such that the combination of all the solutions of the subproblems together approximate the solution of the original control problem. In the framework of [122], each piece of infrastructure is represented by an agent that tries to attain its local objective in close cooperation with other agents.

For a local intersection controller, a new look-ahead traffic-adaptive control approach

is proposed. Currently, most of the control approaches decide control actions based on stages, i.e. they optimize the green times for the pre-defined stages. But the new approach decides the control actions based on the predicted movements of the individual vehicles arriving at the intersection. The movement-based approach is more flexible than the stage-based approach as it allows green for signals in different phases to start sooner if demand for all conflicting movements in the current phase has cleared.

The coordination procedures of the agents are developed on the basis of the actions of nearby agents for two levels, i.e. the microscopic level and the macroscopic level. In the microscopic coordination level, local controllers exchange information of arriving vehicles. When a conflict is anticipated based on the information from neighborhood controllers, the corresponding optimal solution will be iteratively derived by the local controllers. In the macroscopic coordination level, in order to prevent that traffic breaks down on vulnerable parts of the network, the inflow of traffic toward these parts of the network should be constrained. The coordination procedure developed employs two types of constraints: (1) hard constraints enforcing that the volume of traffic entering the vulnerable area does not exceed the volume the infrastructure is able to handle, and (2) soft constraints used to tempt agents further upstream in the network to steer traffic away from the vulnerable area, so as to alleviate the stress put on downstream agents that have already started gating. The microscopic coordination procedure is able to adapt to different traffic volumes and platoon ratios, and to create and to dissolve progression between consecutive intersections. The macroscopic coordination procedure can be used to coordinate all kinds of traffic control measures (e.g. traffic lights, ramp-metering installations, DRIPs (Dynamical Route Information Panels), etc.) at the level of capacities and flows.

Both the distributed and the hierarchical structures can be chosen to reduce the on-line computational complexity, and to make the centralized control system more scalable. The hierarchical control system can be made up of multi-agents (controller) belonging to different levels, as Fig. 2.5 shows. Each agent is supposed to fulfill its predefined task and can make its own control decisions. Agents communicate with each other by transferring information. Therefore, the computational complexity of the centralized controller is separated into multiple small subproblems. A test bed for multi-agent control systems in road traffic management is developed in [123]. Such a test bed can be used to assess different strategies for the application of multi-agent systems for dynamic traffic management and to examine their applicability. It facilitates the development of multi-agent systems for dynamic traffic management.

### 2.2.3 Rule-based strategies

Rule-based systems solve a problem using “if-then” rules [57, 113]. These rules are constructed using expert knowledge and stored in an inference engine. The inference engine has an internal memory that stores rules and information about the problem, a pattern matcher, and a rule applier. The pattern matcher searches through the memory to decide which rules are suitable for the problem, and next the rule applier chooses the rule to apply. These systems are suited to solve problems where experts can make confident decisions. However,

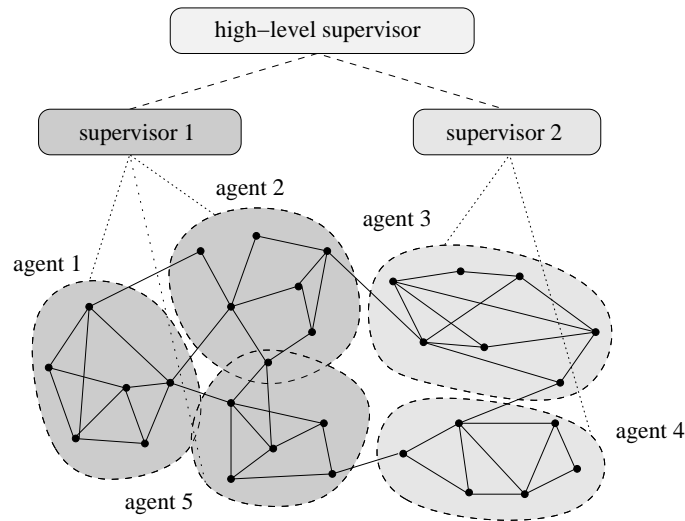


Figure 2.5: Illustration of the coordination of the hierarchical multi-agent system [123]

these systems work only with already created rules and in their basic implementation do not involve learning.

## Freeway networks

### HERO

When the congestion is imported from downstream, local ramp metering almost has no effect. To this end, coordinated control strategies are needed. HERO (HEuristic Ramp-metering coOrdination) [105] is a simple rule-based coordinated ramp-metering strategy that applies ALINEA for the local regulators. HERO can coordinate freeway networks of arbitrary size, including a string containing a number of successive ramps.

The main coordination principle for a string is as follows: Receive the real-time detected ramp queue lengths and mainstream densities from the local controllers; Check whether the relative ramp queue length  $\omega/\omega_{\max}$  exceeds a certain activation threshold, and whether the merge density is close to (or higher than) the critical density. If both conditions are satisfied, this ramp is defined as the master ramp, where the queue may soon reach the maximum admissible value, and then congestion may happen. Therefore, the coordinated control strategy needs to be activated. In order to prevent congestion at the master ramp, coordinated control actions are adopted at the slave ramps (the upstream ramps). Define a minimum ramp queue length  $\omega_{\min}$  for the slave ramps, which is updated according to the real-time changing of the master ramp queue length. By real-time adjusting  $\omega_{\min}$  according to the traffic state of the master ramp, the queue lengths of the slave ramps are increased to stay close to the queue length of the master ramp. In this coordinated control algorithm, the slave ramps hold back some traffic so as to release the pressure from the master ramp, and prevent congestion. When the relative queue of the master ramp decreases below a certain threshold or the mainstream density becomes clearly undercritical, the coordination stops.

The ALINEA-based HERO is shown to outperform the uncoordinated local ramp metering and approximate the efficiency of the sophisticated optimal control schemes (e.g. AMOC) without the effort for real-time modeling calculations or external disturbance prediction [105].

In Netherlands, the HERO algorithm is also applied to the Amsterdam beltway network, but with RWS controller as local controllers. The RWS control strategy, developed by the Dutch Ministry of Transport, is derived from the demand-capacity strategy, which is a feed-forward control approach based on the measured traffic demand on the freeway and the pre-defined capacity of the freeway [117].

### **ACCEZZ**

Fuzzy logic systems, like humans, can handle situations where the available information about the system is vague or imprecise [67, 100]. To deal with such situations, fuzzy sets are used to qualify the variables of the system in a non-quantitative way. Fuzzy sets are characterized using membership functions (e.g. Gaussian, triangle, or normal) that take a value between 0 and 1, and that indicate to what degree a given element belongs to the set (e.g., a speed could be 60 % “high” and 40 % “medium”). The membership degrees can then be used to combine various rules and to derive conclusions. This process consists of three parts: fuzzification, inference, and defuzzification. Fuzzification involves the transformation of a value of a variable into a fuzzy value, by linking it to a given fuzzy set and determining a value for degree of membership. Inference uses a set of rules based on expert opinions and system knowledge and combines them using fuzzy set operators such as complement, intersection, and union of sets. Defuzzification converts the fuzzy output of the inference step in to a crisp value using techniques such as maximum, mean-of-maxima, and centroid defuzzification. One of main difficulties of a fuzzy system can be the selection of appropriate membership functions for the input and output variables. Moreover, fuzzy systems are often combined with other AI techniques for their complete deployment.

As indicated before fuzzy systems can be used when accurate information of the traffic model is difficult to obtain or is not available [15, 74]. A fuzzy logic controller for ramp metering with a description of the various steps (fuzzification, inference, and defuzzification) is presented in [126]. Several fuzzy sets that can relate a variable (input, output) to a particular situation can be defined such as fuzzy sets for local speed, local traffic flow, queue occupancy, metering rate, and local occupancy. Using fuzzification input variables such as speeds, flows, occupancy levels in the vicinity of the fuzzy ramp meter controller and output variables such as metering rates can be translated to fit the defined fuzzy sets and to obtain values for the degree of membership. Next, these values are fed to the inference engine, which is constructed using a set of rules based on the experience of traffic control center operators and on off-line simulations. The result of the inference is then transformed into a crisp value in the defuzzification step, after which the final result is applied to the traffic system or presented to the operator of the traffic control center for further assessment.

ACCEZZ (Adaptive and Coordinated Control of Entrance Ramps with Fuzzy Logic) [16] is a rule-based algorithm for coordinated ramp metering. The rule base is defined as a set of rules in fuzzy logic, incorporated with human expertise. Fuzzy logic allows simple development and modification, because rules are easy to define, alter, or eliminate. The fuzzy controller also compensates for poor, inaccurate measurements. Since a fuzzy con-

troller can easily handle nonlinear systems with unknown models, it has a distinct advantage for controlling complex traffic systems.

The ACCEZZ approach can be developed into several versions of the algorithms, i.e. Neuro-Fuzzy Online, Neuro-Fuzzy Offline, Genetic Fuzzy Online, Genetic Fuzzy Offline, and Genetic Fuzzy Reality. The core of the ACCEZZ model family is a fuzzy controller (see Fig. 2.6). The control rules are expressed by defining a number of fuzzy sets that are identified and derived from heuristics, expert knowledge, and simulation testing. The inputs of the fuzzy controller are measured on the mainlines and on-ramps, i.e. local speed, local traffic flow, and local occupancy, upstream and downstream of the on-ramp. The output of the fuzzy controller, the metering rate, is calculated every minute based on these real-time measured and historical traffic data.

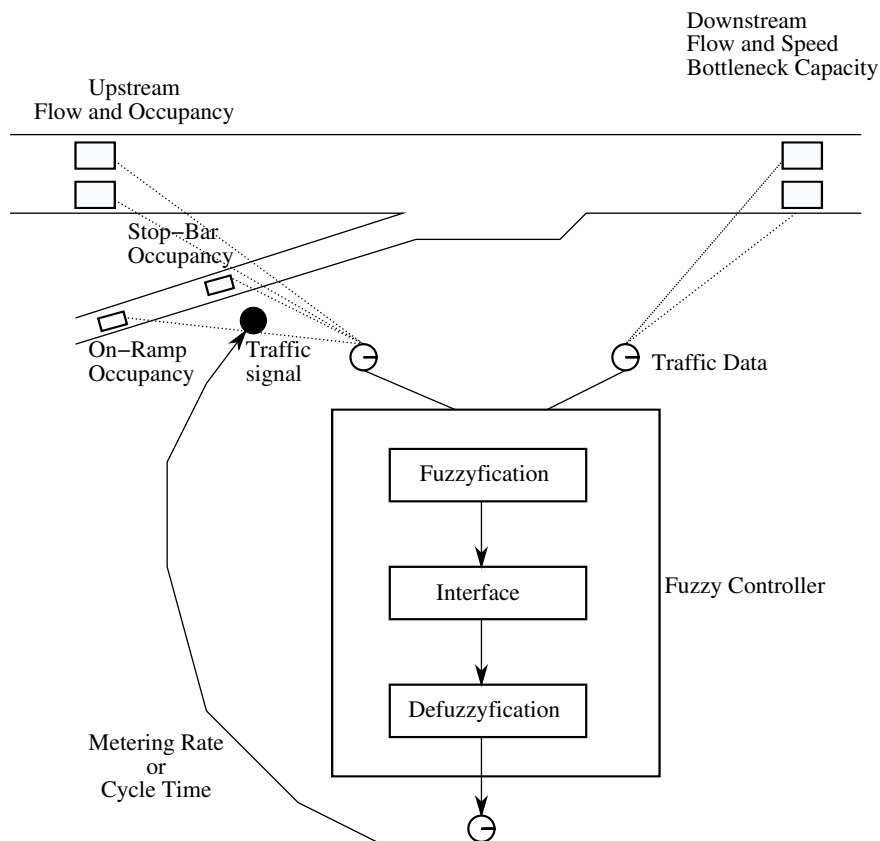


Figure 2.6: Fuzzy ramp metering [16]

In order to coordinate the local fuzzy ramp metering controllers, the shape of each input or output fuzzy set at each on-ramp location of the metered freeway is adjusted dynamically. So, one way of modifying the behavior of the ramp metering algorithm is by recalibrating the parameters of each fuzzy set, i.e. redefining the linguistic variables. Learning/optimization methods obtained from neural network theory or evolutionary algorithms are used to find the optimal parameters of the fuzzy sets. The neuro-fuzzy algorithm learns



the fuzzy parameters aiming at minimizing the Total Time Spent in the metered freeway system. The macroscopic traffic model METANET is used to evaluate the different coordinated ramp metering strategies, and helps to find the best system-wide strategy. Alternatively, a genetic algorithm can be used to determine the optimal coordinated parameters of the fuzzy ramp metering controllers based on macroscopic traffic model METANET. The resulting systems are either called neuro-fuzzy or genetic fuzzy ramp metering.

Comparing with five other standard ramp metering algorithms, i.e. demand-capacity, occupancy strategy, ALINEA, Denver's HELPER algorithm, and Minnesota's Zone approach, all developed versions of the ACCEZZ model family substantially improve the traffic conditions for the freeway analyzed [16].

## Urban networks

### Urban traffic control based on hybrid petri nets

An urban network of signalized intersections can be suitably modeled as a hybrid system<sup>1</sup>, in which the vehicle flow behavior is described by means of a time-driven model and the traffic light dynamics are represented by a discrete-event model. Petri nets are known to be very suitable to model discrete-event systems, since they are able to capture the precedence relations and interactions among the concurrent and asynchronous events that are typical of discrete-event systems. A Petri net is a directed bipartite graph, in which the transitions (i.e. events that may occur) are signified by bars and the places (i.e. conditions) are signified by circles. Based on the hybrid Petri net model, a rule-based urban traffic control structure [39] (see Fig. 2.7) is proposed to control and coordinate traffic networks, aiming at improving the performance of some classes of special vehicles, i.e. public and emergency vehicles.

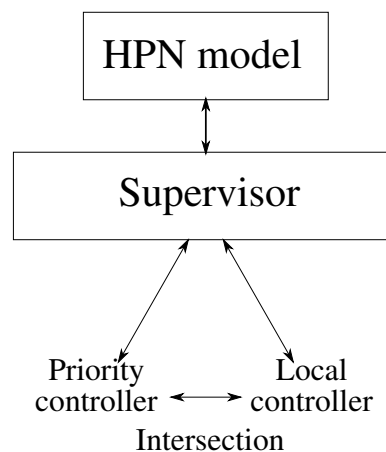


Figure 2.7: Control structure based on a hybrid Petri net model (adopted from [39])

Traffic lights can be optimized under both regular and special conditions with this con-

<sup>1</sup>Hybrid systems are systems that are characterized by both continuous and discrete-event dynamics (switching). They exhibit both continuous dynamics (which can be modeled by differential or difference equations) and discrete-event behavior (switching).

trol structure. That is to say with or without a public or emergency vehicle asking for a privilege. Fig. 2.7 illustrates the operation of the system. 1) The local controller regulates the intersection under the hypothesis of regular conditions, by optimizing the traffic light plan based on the traffic model; 2) The priority controller is actuated to force a modified traffic light plan when a particular event occurs; 3) The supervisor coordinates all the local and priority controllers and solves all the problems involving several intersections.

The local controller executes a traffic-responsive plan that optimizes the phase splits taking the queue length at the intersection as the performance. The priority control rule at a intersection is: 1) If a privileged vehicle asks for priority, then the arriving time of the vehicle will be predicted by the hybrid traffic model; 2) If the traffic light is green, when the vehicle arrives, then there will be no intervention (i.e. the algorithm stops); 3) If a red light is expected, the local priority controller has to decide whether to extend the current green time, or to stop the current red light earlier, and to anticipate the next green time.

The rule-based urban traffic coordinated control structure via hybrid Petri nets is able to take public and emergency vehicles into consideration. The hybrid Petri net model used has been validated through real traffic data about the Italian city of Torino [39].

### **Fuzzy rule control system**

Similar to ACCEZZ for freeway networks, fuzzy-logic controllers with genetic algorithms or neural network algorithms as adapting approaches for the fuzzy rules are also applied in urban traffic systems.

In [61], a decentralized urban traffic structure is proposed. It applies a fuzzy-logic controller as local intersection controller, and a dynamic-programming<sup>2</sup> technique to coordinate the control results obtained from fuzzy-logic controllers and to derive the green time for each phase in a traffic-light cycle. In each fuzzy-logic controller, a GA algorithm is applied to learn and update in real-time the fuzzy sets.

A more complex urban network control hierarchical architecture is given in [26] based on a fuzzy neural decision support principle, as Fig. 2.8 shows. The architecture consists of three layers. The lowest layer consists of intersection controller agents that control individual, preassigned intersections in the traffic network. The middle layer consists of zone controller agents that control several preassigned intersection controller agents. The highest level consists of one regional controller agent controlling all the zone controller agents.

In each layer, every agent can obtain traffic data and make decisions autonomously. Both lower layer agents and upper layer agents can send cooperative factors (requests) to each other. In the zone controller agents, fuzzy control algorithms are adapted by changing the fuzzy rules using evolutionary algorithm, i.e. neural network algorithm. Several techniques, including reinforcement learning, weight adjustment, and adjustment of fuzzy relations, have been applied to adapt the dynamics of the agents online.

---

<sup>2</sup>The basic idea behind dynamic programming is to decompose a multistage decision problem into a number of subproblems that calculate the optimal path between all states before and after one decision stage.

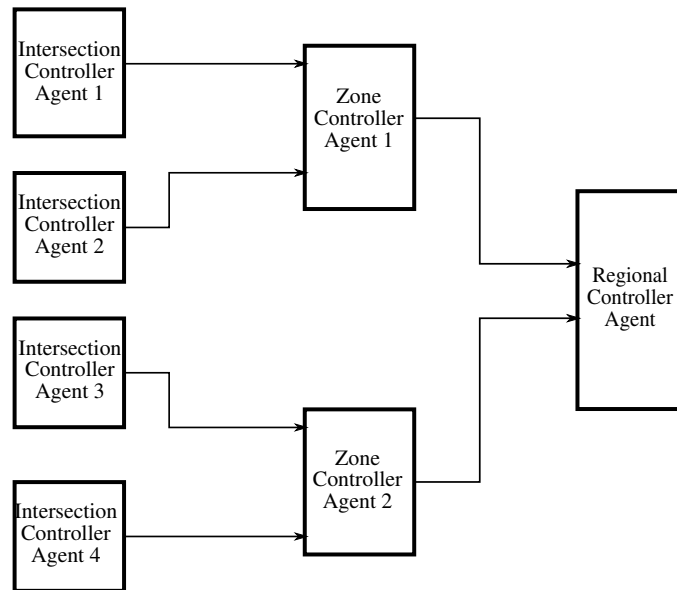


Figure 2.8: Multi-agent architecture for urban traffic signal control (adopted from [26])

## Mixed freeway and urban networks

### HARS

HARS (Het Alkmaar RegelSysteem) [75, 124, 125], which means “The Alkmaar Control System”, is a state-of-the-art traffic management system implemented in the Alkmaar region in the Netherlands. The HARS system combines both a top-down traffic management strategy and a bottom-up traffic management strategy into a hierarchical traffic network management architecture. The two traffic management strategies complement each other. The top-down strategy makes decisions on the control schemes based on the predefined traffic scenarios stored in the expert database. In order to overcome some drawbacks of the top-down structure, the HARS system adopts an agent-based bottom-up traffic management architecture. In the bottom-up strategy, all road segments and nodes that connect the segments are defined as agents. The agents can communicate with each other, and coordinate with each other to make control decisions according to the predefined rules based on the expert knowledge.

Fig. 2.9 shows the structure of the HARS system. In the “Data gathering” block, the real-time traffic information (traffic states) are measured and collected through the loop detectors. Based on these measurements, the traffic model MaDAM [11] is used to determine what the traffic state is on links that have no sensors of their own in the “Traffic model” block. In addition, MaDAM predicts what the traffic states will be for the links in the next 30 minutes in blocks of 5 minutes. In the “Network management” block, both the agent controllers and the expert control schemes are implemented. For the top-down expert controller, proper control strategies are chosen based on the identified scenario. For the agent controllers, link agents compare the information (obtained from the “Traffic model” block)

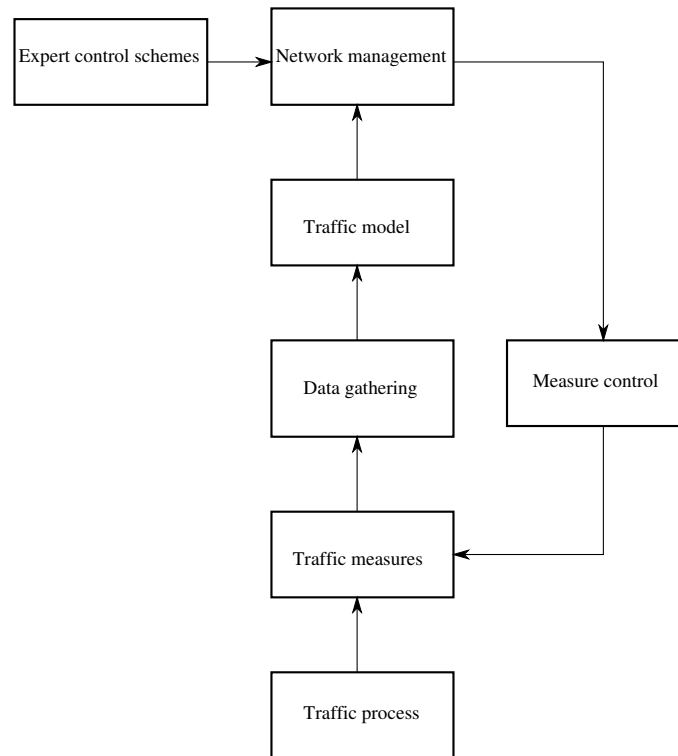


Figure 2.9: The structure of the HARS system (adopted from [75])

about their traffic state with a so called reference framework. The reference framework defines criteria that the traffic state on the link should meet. If the link's traffic state deviates from the reference framework, or will deviate in the near future, links will communicate via intermediate nodes with other links and ask them to reduce outflow in order to meet the criteria. If the upstream link is not able to adjust its outflow to make the downstream link meet the criteria, then it will forward the service-call to its upstream neighboring link(s). The derived coordinated control actions are sent back to the traffic process to be implemented. All the predefined configurations of the control strategy and the reference frameworks are stored in a database, represented by the "Expert control schemes" block. The "Measure control" block represents the local controller, which is taken as backup controller in case that the coordinated controller fails.

Alkmaar has two types of control measures: traffic light systems and Dynamic Route Information Panels (DRIPs). The DRIPs will be used for rerouting and informing drivers. The traffic light systems will be used as an instrument to change intensities of traffic flows.

#### 2.2.4 Case-based strategies

Case-based reasoning, as the name suggests, solves a problem using the knowledge that was gained from previously experienced similar situations (cases) [1, 111]. In this way, this

technique learns the way a new problem is solved and stores the new solution in a database. A disadvantage of this approach is that it might not be clear what should be done for a case that is not yet present in the case base. However, new cases could be added on-line to deal with this problem.

Dynamic traffic management focuses on the integrated (as opposed to isolated) and coordinated (combination of different measures, e.g. ramp-metering, variable speed limits, dynamic route guidance, opening shoulder lanes, providing route information, etc.) deployment of measures, anticipating on future changes in traffic conditions. In the regional traffic management centers, traffic operators decide when and which dynamic traffic management measures are to be deployed in case of recurrent and non-recurrent conditions. To improve the existing dynamic traffic management systems, BSES (Boss Scenario Evaluation System) [38, 64, 65] based on fuzzy multi-agent case-based reasoning is proposed.

BSES can evaluate control scenarios in real time, predicting their effects in terms of various measures of effectiveness, such as total travel time, vehicle loss hours, average speeds, fuel consumption, etc. The main characteristics of the system are 1) that it is case-based, i.e., it uses either synthetic or real-life examples of the effect of control scenarios under different circumstances; 2) that it determines the similarity of the current situation to different examples in the case base using fuzzy logic, and 3) that it is agent-based, meaning that it predicts the effects of the different measures for small subnetworks and combines these predictions afterwards.

Due to the exponential growth of the case base, straightforward application of case-based reasoning to the decision support task is not feasible. Therefore, representative cases that can occur in practical situations are required to find out first how to reduce the case base scale. To address this problem, two aspects are introduced into the case-based reasoning framework: 1) Fuzzy logic is used to combine different cases in the case base (fuzzy case-based reasoning); 2) The network to be controlled is divided in  $n$  partially independent subnetworks for which the aforementioned fuzzy case-based reasoning approach can be applied. An iterative approach is used to find consistent solutions for the subnetworks.

The main advantages of the BSES approach are the speed of computation (compared to using traffic flow models), the ability to use actual knowledge directly (rather than general knowledge or simulated data), and the ability to learn from previous experiences (continuous step-wise learning). It turns out that the system is able to very quickly produce predictions on the impact of different control scenarios to the traffic operations in the network, and that it can thus support operators in their decision tasks in a real-time decision environment.

### **2.2.5 Anticipatory control strategies integrated with traffic assignment**

Traffic control discussed here generally refers to controlling the traffic control measures (e.g. traffic lights, traffic information, and ramp-metering) to reduce the traffic delay in the traffic network. However, the travelers inside the network may change their routes, when the new traffic control measures change the traffic in the network. Therefore, traffic control and the behavior of the travelers influence each other. As a result, a new traffic control strategy is constructed by combining the traffic control problem with the traffic assignment problem. The new traffic control problem is formulated into a bi-level program in which the upper level deals with the control problem, and the lower level with the assignment problem.

In [116], an anticipatory control strategy is proposed to control and coordinate urban

traffic networks by predicting the future traffic flows within the network taking the variation of the traffic assignment into consideration. As the traffic control and the behavior of the travelers have different goals, game theory is applied to solve the bi-level optimization problem of the anticipatory control. The traffic control engineer and all the road users are then considered as two players. The traffic engineer controls the signal settings and the road users have route choice. A Nash game is played when both players react on each other's moves: the traffic engineer sets the signal control plans, the road users travel and select routes based on their individual preferences and the experienced travel times. The game ends when reaching the Nash equilibrium, which is the situation when no player can benefit by changing his strategy, while the others keep their strategies unchanged. In every iteration, an optimization problem is solved to obtain the best control plan for the predicted time period and for the whole traffic network in the upper level. Then, the road users chose their routes according to the travel costs under the new plan. The traffic engineer decides on a signal plan based on the anticipated traffic flows and the road users react by changing their routes, and this procedures repeats until an equilibrium is reached.

Fig. 2.10 illustrates the framework for developing, testing, and evaluating all kinds of network control strategies. The “optimization control plan” is the part where the anticipatory control strategy is determined. After the control plan is derived by certain algorithm, a simulation is started with a dynamic network loading model to see how traffic propagates through the network with this control plan; based on these results a dynamic traffic assignment is run to obtain a new route flow distribution, and again the dynamic network loading model is run to come to a final evaluation of the control plan.

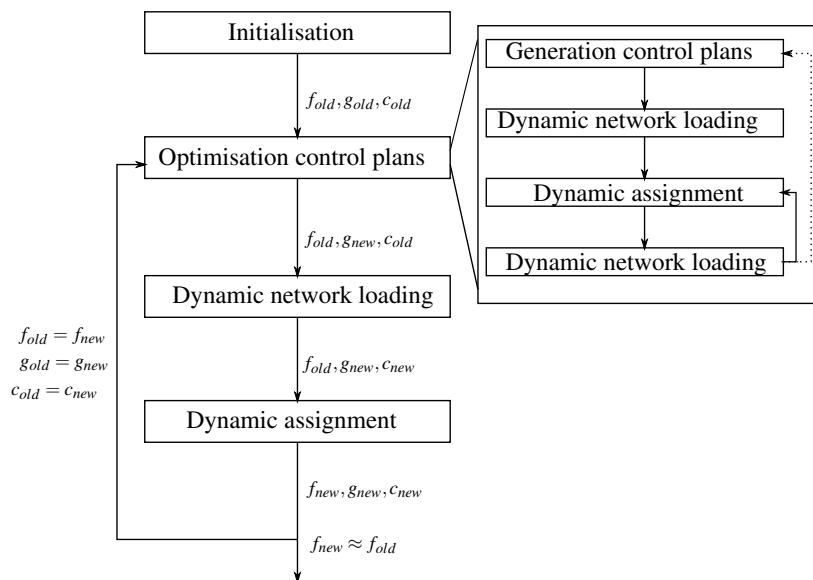


Figure 2.10: Framework for anticipatory control (adopted from [116])

### 2.2.6 Alternative approaches

According to [35], a Macroscopic Fundamental Diagram (MFD) for urban traffic streets is proved to exist, and [53] shows that an MFD also exists on a neighborhood-sized sections of cities, and that it is independent of the demand. This result is tested in neighborhoods of the order of  $10 \text{ km}^2$  in cities like Yokohama, Japan, etc. The experiment data are gathered by GPS-equipped taxis and fixed detectors. The network MFD obtained is illustrated in Fig. 2.11. It can be used to control network demands to improve accessibility. Simple versions of the control strategies based on the network MFD are already used in London, Stockholm, and Singapore, etc.

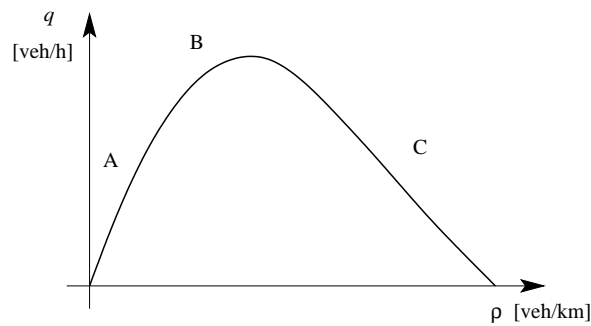


Figure 2.11: Network MFD

A large traffic network can be divided into several small neighborhood subnetworks. The subnetworks are controlled and coordinated so that they remain in Region B on the MFD (see Fig. 2.11), in order to maximize throughput of the networks. Based on the MFD of the subnetworks, a series of coordination approaches can be developed:

- Rule-based control
- Proportional control
- Anticipative control

**Rule-based control** As Fig. 2.12 shows, the network is divided into 5 subnetworks with different subnetwork MFDs. We can see that subnetwork 4 is congested. The coordination control rules can be designed as: 1) Check the neighboring subnetwork status of the problem subnetwork (i.e. subnetwork 4), to see whether they are in the safe regions (Region A and Region B) on the MFDs (as subnetwork 2 and 5 are in safe regions). 2) Alleviate the congestion by reducing the output flow of the neighboring subnetworks from the subnetwork with the lowest priority on until the problem is solved, i.e. first reduce the flow from subnetwork 5 to subnetwork 4.

**Proportional control** Proportional control is applied to control network traffic flow in the Region B of Fig. 2.11, which takes the real-time detected network traffic states as feedback:

$$Q^*(k+1) = Q(k) + \alpha(N^* - N(k)) \quad (2.3)$$

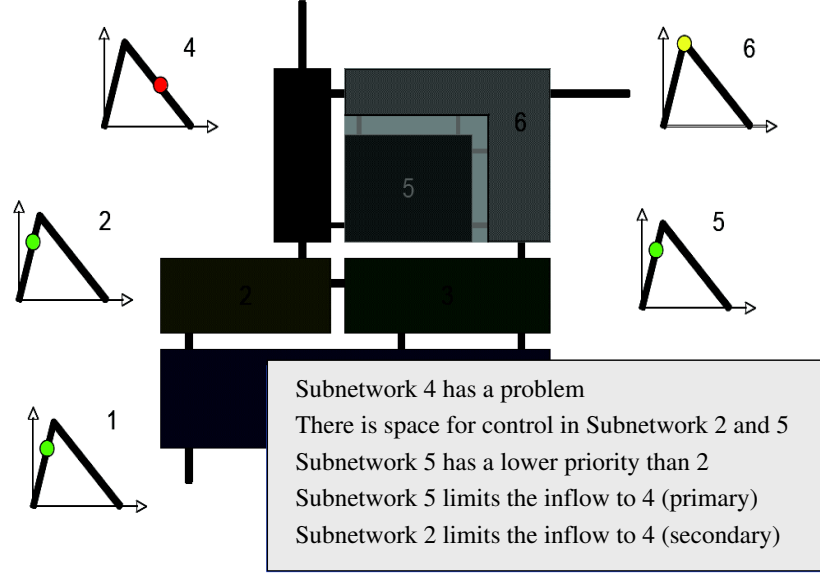


Figure 2.12: Rule-based control strategy based on network MFD

where  $Q(k)$  is the network flow rate at time  $k$ ,  $N(k)$  is the number of vehicles in the network at time  $k$ ,  $N^*$  is the traffic state control set-point derived from the MFD (expressed as the number of vehicles in the network), and  $\alpha$  is the control gain. Proportional control aims at controlling the traffic network within the predefined region (e.g. Region B) by adjusting in real-time the input and output traffic flow of the network.

**Anticipative control** A simple *continuous dynamic* subnetwork model is established based on the network MFD. For subnetwork  $i$ , we have

$$\frac{dN_i}{dt} = \sum_j q_j(t) - Q_i(N_i(t)), \quad (2.4)$$

where  $j \in N_i$  is the neighborhood subnetwork of subnetwork  $i$ ;  $q_j(t)$  is the anticipated output flow rate of subnetwork  $j$  into subnetwork  $i$  at time  $t$ , and  $\sum_j q_j(t)$  is the sum of the input flow rates for subnetwork  $i$ ;  $Q_i(N_i(t))$  is the predicted output flow rate of subnetwork  $i$  derived from the MFD function based on the current network traffic state  $N_i(t)$  (i.e. the number of vehicles). This model can roughly describe and predict the traffic states evolution of the subnetworks, which are influenced by the traffic flow exchange among subnetworks. An anticipative control strategy can be derived to coordinate the subnetworks based on this simple model.



## 2.3 String coordination approaches

A string is a link in the freeway network, or a road in an urban network. It belongs to a freeway network or a urban traffic network, and it generally plays an important role in the traffic networks. Therefore, some research has been developed focusing on coordination of strings. As a part of the traffic network, almost all the coordination strategies mentioned above are also suitable for the string. Moreover, some traffic management systems [48, 97] above are designed taking the string coordinated control objective as the target of one of the layers in the hierarchical control structure. Nevertheless, there are also some approaches that focus explicitly on strings, and they will be discussed next.

### Freeway strings

The cell transmission model [33, 77] divides a freeway into  $N$  sections or cells, each with one on-ramp and one off-ramp (see Fig. 2.13). In the figure, vehicles move from the right to the left. Section  $i$  is upstream of section  $i - 1$ . There are two boundary conditions: free flow prevails downstream towards Section 0, and on the upstream of the freeway here is an on-ramp with an inflow of  $r_N$ . The flow accepted by Section  $N - 1$  is  $f_N(k)$  vehicles per period at time step  $k$ . The cumulative difference leads to a queue of size  $n_N(k)$  in period  $k$ , and  $r_i$  and  $s_i$  are on-ramp flow and off-ramp flow for Section  $i$  at time step  $k$  respectively. The density  $n_i$  in Section  $i$  is updated as

$$n_i(k+1) = n_i(k) - f_i(k)/(1 - \beta_i) + f_{i+1}(k) + r_i(k), \quad 0 \leq i \leq N - 1 \quad (2.5)$$

where  $\beta_i$  is the split ratio.

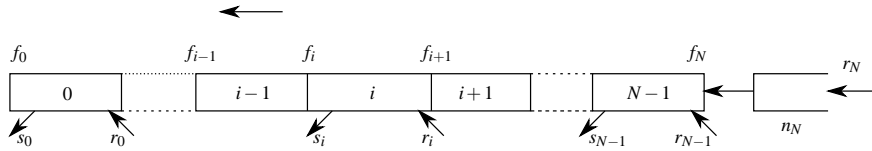


Figure 2.13: The freeway has  $N$  sections. Each section has one on- and one off-ramp. (adopted from [55])

Reference [55] provides a complete analysis of the behavior of the cell transmission model of a freeway with stationary demand. The state of the dynamical system is the  $N$ -dimensional vector  $n$  of vehicle densities in the  $N$  sections. The key to the behavior of cell transmission model is the location of bottleneck sections where flow equals capacity. The bottlenecks partition the freeway into decoupled segments. Each decoupled segment starts with a bottleneck and ends just before the next upstream bottleneck. In each segment, the equilibrium set is determined only on its own condition, and the number of congested sections in the segment depends on which equilibrium in the set the segment belongs to. Each equilibrium is stable and every dynamic trajectory of the traffic states converges to certain equilibrium state. Let  $r$  be a demand vector for all segments and  $\phi$  the resulting equilibrium flow vector for all segments, then

$$\phi_N = r_N, \quad \phi_i = (1 - \beta_i)(\phi_{i+1} + r_i), \quad 0 \leq i \leq N - 1, \quad (2.6)$$

Table 2.1: Comparison of the features for different traffic coordination control methodologies

Methodologies		Information	Control quality	Complexity	Integrated	Scalable	Application effort
MPC	Centralized	Global	H+	H	Yes	No	H
	Hierarchical	Compromise	H	H	Yes	Yes	H
	Distributed	Local	H	H	Yes	Yes	H
Optimal control		Global	M	H	Yes	No	H
Rule-based		Compromise	M+	L	Potentially	Yes	M
Case-based		Compromise	M+	M	Yes	Yes	M
Anticipatory control		Global	H+	H	Yes	No	H
MFD-based		Global	M-	L	Yes	Yes	L

H - High; M - Medium; L - Low; Compromise - between global and local

which means that the equilibrium flows in the segments depend on both the traffic demands and the current traffic flow states for the segments.

It is proved that, under the following two conditions: 1) when bottleneck is caused by demand exceeding the capacity; 2) when congestion already exists as initial condition, ramp metering control can avoid congestion from happening or relieve congestion that already exists.

### Urban strings

Urban string control mainly refers to arterial progression coordination, which maximizes the green traffic light band to reduce the traffic delays on the arteries. In [51], a mathematical programming model for the development of optimal arterial-based progression schemes is proposed. Under such a scheme, a continuous green band is provided in each direction along the artery at the desired speed of travel to facilitate the movement of the principal through-flows along the arterial. Both uniform and variable bandwidth models are formulated. New approaches generate variable bandwidth progressions in which each directional road section is allocated an individually weighted band that can be adapted to the prevailing traffic flows on that link. Mixed-integer linear programming is used for the optimization. Simulation results indicate that this method can produce considerable gains in performance when compared with traditional progression methods. A real-time progression optimization approach can be also found in [50], which is a multi-level real-time traffic-adaptive control algorithm taking the dynamic traffic assignment and the routing capacity into consideration.

## 2.4 Comparison

The main characteristics of the methods discussed above are summarized in Table 2.1 and Table 2.2. As Table 2.1 illustrates, in general, the more elaborate information that the controller takes into consideration, the better control result will be obtained. The centralized MPC approach makes use of the total global information by applying a traffic network model and feeding the model with real-time detected traffic states. So, the centralized MPC approach has the highest coordination control quality. However, it also has a high computational complexity at the same time, and needs more efforts to implement. Therefore, distributed and hierarchical MPC structures are developed to solve this problem by making some compromises. They give up a part of the global information to obtain simplified subproblems, and improve the applicability of the approaches by controlling and coordinating the subproblems. Moreover, a distributed structure also makes the controller scalable. Despite of the drawback of the high computational complexity, the centralized MPC approach still has the best global control performance, and can be used for long-term traffic control or planning, in which the control algorithm does not need to react very fast. Other solutions to this problem are to use distributed and hierarchical MPC, or to develop efficient model predictive control algorithms for large traffic networks that are applicable to real-life traffic, and keep good control performance as well.

Rule-based and case-based approaches are control strategies mostly based on historical information and expert experience. Because they are comparatively easy to implement, simple rule-based and case-based approaches first have been applied in traffic management system to coordinate traffic networks at the beginning. Moreover, they are the control approaches that are easy to coordinate all kinds of traffic control measures and manage large complex transportation systems. However, the control plans obtained by rule-based and case-based strategies are in general not optimal solutions. But, some smart rule-based and case-based control systems (e.g. HERO, ACCEZZ, HARS, BSES, etc.) can adjust themselves by updating their rules or databases according to the real-time measured traffic states or the predicted traffic states through the traffic models. This makes the rule-based and case-based approaches more adaptive to the variation of the real traffic.

In fact, when the traffic control plans change, the traffic flows in the traffic network will be reassigned, because the road users will also change their routes. Therefore, it is more realistic to also consider the traffic assignment while controlling the traffic. The anticipatory control approach constructs a bi-level program problem, in which the upper level deals with the control problem, and the lower level with the assignment problem. The control results of the anticipatory control are good because of taking the dynamic traffic assignment information into consideration. However, because of the iterative feature of the solver, the anticipatory control approach suffers the same drawback as the MPC control approaches, i.e. high computational complexity. Just like the centralized MPC, the anticipatory control can also be used for long-term traffic control and planning.

Recently, the Macroscopic Fundamental Diagram (MFD) has also been discovered to exist for a neighborhood-sized traffic network. The family of network MFD-based approaches to coordinate traffic subnetworks is scalable, the easiest strategy to be implemented, and they have a very low computational complexity. Of course, they also result in a rougher approach, which cannot guarantee a very high control quality.

## 2.5 Summary

Traffic states and traffic control measures are not isolated, they interact with each other. Therefore, coordinated traffic control strategies are very important for coordinating all the traffic control measures to improve the transportation environment.

In this literature survey, coordinated traffic control strategies for both traffic networks and strings are summarized. As a part of the traffic network, traffic strings can also be controlled and coordinated by most of the strategies for traffic networks. From the view of traffic control methodologies, the existing coordinated traffic control strategies can be classified into MFD-based (Macroscopic Fundamental Diagram based) approaches, case-based approaches, rule-based approaches, anticipatory control approaches, optimal control approaches, and MPC (Model Predictive Control) approaches under centralized, distributed, and hierarchical control structures. The characteristics of these methodologies have been analyzed and compared in this literature survey.

Anticipatory control, optimal control, and MPC all belong to the same category, advanced model-based optimization control methods. The family of model-based optimization control approaches is the most powerful strategy, as it has high control quality and is capable of coordinating all integrated control measures at the same time. However, the drawbacks of model-based optimization approaches are: they have a comparatively high real-time computational complexity, and it takes more effort to be implemented and get it working in real-life traffic applications.

On the other hand, the MFD-based approaches are scalable, the easiest strategies to be implemented, and have a very low computational complexity. Of course, they are also rougher approaches that cannot guarantee a very high control quality. However, at present the network MFD-based control approaches are methods that can be implemented in practice and start working in a short time period.

Compared with the former categories of control methodologies, rule-based approaches and case-based approaches seem to offer a good trade-off between control performance and complexity.

In this chapter, we provide a general summary on the state-of-art of the existing coordinated traffic control approaches. In the following of the thesis, we will focus on a specific layer of coordinated traffic control problems for urban traffic networks. To achieve higher control quality in the future, more advanced traffic coordinated control approaches need to be considered. MPC approaches are promising methods that can provide a good global coordination performance for traffic networks. But more work is necessary to improve the efficiency of MPC controllers for traffic networks to make it applicable in practice, and to build user-friendly interface to make MPC easier to be implemented in reality. Therefore, the emphasis of the thesis is mainly put on investigating efficient model predictive control methods for large-scale urban traffic networks.

Table 2.2: Comparison of the coordinated traffic control approaches

Approaches			Control field	Control Feature						Control Structure			Control Measure				Control Objective		
				Model-based	Rule-based	Case-based	Optimization	Feed back	Prediction	Centralized	Distributed	Hierarchical	Ramp metering	Speed limits	Route guidance	Traffic signals	TTS like	Travel time	Delay
MPC	Centralized	Hegy et al.(2005)[59]	F	✓			✓	✓	✓	✓			✓	✓					
		van de Berg et al.(2004)[119]	U/I	✓			✓	✓	✓	✓				✓	✓				
		Dotoli et al.(2006)[42]	U	✓			✓	✓	✓	✓				✓	✓				
	Hierarchical	van Katwijk et al.(2008)[122]	U	✓			✓	✓	✓			✓		✓	✓				✓
		Kotsialos et al.(2005)[73]	F	✓			✓	✓	✓			✓			✓				
		OPAC[48]	U	✓			✓	✓	✓			✓			✓				
		RHODES[97]	U	✓			✓	✓	✓			✓			✓				
	Distributed	Ghods et al.(2008)[54]	F	✓			✓	✓	✓		✓		✓	✓					
CoSIGH[25]		U	✓			✓		✓		✓				✓		✓			
Optimal Control	AMOC[70]	F	✓			✓		✓	✓			✓		✓					
	OASIS	F	✓			✓		✓	✓			✓		✓					
	QPC[3]	U	✓			✓		✓	✓					✓					
	NOC[3]	U	✓			✓		✓	✓					✓					
	TUC[3]	U	✓			✓	✓	✓	✓					✓					
	DISCO[90]	U	✓			✓		✓	✓					✓				✓	
Rule-based	HERO[105]	F		✓			✓				✓	✓							
	ACCEZZ[16]	F	✓	✓		✓					✓	✓			✓				
	HARS[75]	I	✓	✓				✓			✓	✓		✓					
	Febbraro et al.(2004)[39]	U	✓	✓		✓	✓	✓			✓			✓					
	Heung et al.(2005)[61]	U		✓		✓					✓			✓				✓	
	Choy et al.(2003)[26]	U		✓							✓			✓				✓	
Case-based	BSES[38]	F			✓						✓	✓							
Anticipatory Control	Taale et al.(2008)[116]	I	✓			✓	✓	✓			✓	✓		✓		✓		✓	
MFD-based	Rule-based	U		✓										✓					
	Proportion control	U					✓							✓					
	Anticipation control	U	✓			✓	✓	✓						✓					

F - Freeway; U - Urban; I - Integration of freeway and urban



## Chapter 3

# Framework for MPC Control of Large-scale Urban Traffic Networks

Traffic congestion is a serious issue for urban areas, especially for big cities. Because the population density in large cities is particularly high, and thus the requirements for transportation are also high for both commercial reasons and personal reasons. Moreover, urban road networks grow larger and larger, and become more and more complex. Therefore, an efficient transportation management system is necessary. Thus, a traffic control system based on Model Predictive Control (MPC) is proposed for urban traffic. To better control a large-scale urban road network, a proper framework of traffic controllers is necessary for the following reasons:

- **Physical structures:**  
When a city becomes very large, it can be composed of many districts with their own local traffic management centers, or by a downtown area and several satellite cities surrounding the downtown. So, a large-scale urban traffic network can be geographically divided into many subnetworks. Therefore, it is necessary to design a control framework with multiple levels to optimize the overall performance and to coordinate the underlying subnetworks.
- **System dynamics:**  
The dynamics of a traffic system can be either slowly changing or fast changing. For instance, similar traffic flow features will repeat on a weekly basis, while slowly changing during months or even years. But, traffic flow can also change fast within an hour or even a minute. Moreover, the traffic system dynamics can also vary differently for different levels of the traffic network scale. For example, the traffic dynamics of an intersection can be modeled at a faster time scale, while the traffic dynamics at the network level may be described at a slower time scale. Therefore, according to the specific dynamics of traffic systems, a structured traffic control system is necessary aiming at regulating the traffic on different temporal levels.

- **Control objectives:**  
A traffic system is a large complex system composed of freeway transportation, urban transportation, public transportation, pedestrians, etc. The integrated traffic system has different control objectives for different types of transportation. A structured control system can coordinate the multiple control objectives.
- **Control models:**  
The traffic models used for control differ from each other a lot. There are models for a variety of transportation systems, freeways, urban roads, pedestrians, etc. There are also models with different modeling powers, i.e., different modeling details or different levels of descriptive abilities. Some models are more elaborate and complex, while some are rougher and simpler. Due to the differences among traffic models, a distributed control structure can be applied to coordinate, and multiple levels of controllers can be designed based on different models aiming at solving the specified problem.
- **Computation complexity:**  
When the scale of the urban traffic network increases, the computational complexity of a centralized MPC controller grows about exponentially in practice<sup>1</sup>. So the CPU time needed for solving the on-line optimization problem for the centralized MPC controller will become very long, and as a consequence the optimization problem will become real-time infeasible when the MPC controller is implemented in a real-life traffic network. Therefore, the large urban traffic network needs to be divided into several smaller subnetworks so as to divide the computational burdens and then be coordinated by a coordination framework.

## 3.1 Network-wide control structures

When a traffic network becomes large, it is necessary to control the large-scale traffic network under a proper control structure. A well-structured traffic control system can avoid or solve the problems mentioned above, and also may help the system achieve a better performance [114].

### 3.1.1 Centralized control structure

In centralized control, the controlled system is modeled as a whole system, and is controlled by one overall controller (see Fig. 3.1). This controller supervises the whole system, and has full information about the system. Based on the full information, the controller makes the decisions that are optimal for the entire system, and sends them to the system to implement. The centralized controller has full information shared by the entire system, which is called classical information pattern, in which the overall information of the system is centralized and known by every element within the system. For a system with  $s$  elements, the classical

---

<sup>1</sup>The MPC optimization problem is in general non-convex and nonlinear, and thus NP-hard [47]. In practice, this means that the execution time will increase exponentially as the problem size increases.



information pattern can be expressed as

$$\begin{aligned} I(t) &= \{(\mathbf{u}_i, \mathbf{y}_i) | \mathbf{u}_i : [t_0, t] \rightarrow \mathbb{R}^{n_{u,i}} : \tau \mapsto \mathbf{u}_i(\tau), \mathbf{y}_i : [t_0, t] \rightarrow \mathbb{R}^{n_{y,i}} : \tau \mapsto \mathbf{y}_i(\tau), i = 1, 2, \dots, s\} \\ \mathbf{u}(t) &= f(I(t)), \end{aligned} \quad (3.1)$$

where  $\mathbf{u}_i$  and  $\mathbf{y}_i$  are the input and the output signals of element  $i$  in the system, and  $I(t)$  is the overall full information of the system at time instant  $t$ . The full vector of the control inputs to the system,

$$\mathbf{u}(t) = \begin{bmatrix} \mathbf{u}_1(t) \\ \mathbf{u}_2(t) \\ \vdots \\ \mathbf{u}_s(t) \end{bmatrix}, \quad (3.2)$$

depends on the overall information of the system, i.e.  $I(t)$ .

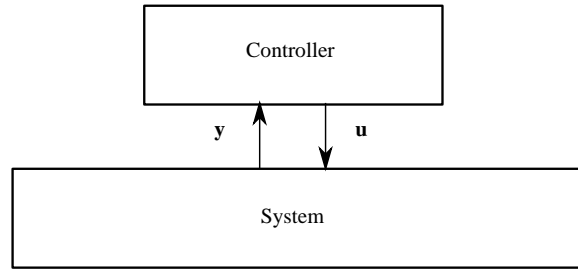


Figure 3.1: Centralized control structure

In principle, if an MPC controller is applied in a centralized structure, it can derive and guarantee the globally optimal control actions for the whole traffic network based on the classical information pattern. It can maximize the throughput of the whole network, and provide network-wide coordination of the traffic control measures. However, the problem is that the on-line computational complexity for centralized MPC grows about exponentially in practice, when the network scale gets larger. Therefore, even though centralized MPC can guarantee the best overall control performance for the whole traffic network in theory, it also pays the highest computational cost for its good performance, and suffers from the risk of not being applicable in real-life traffic.

### 3.1.2 Decentralized control structure

Decentralized control divides the overall system into  $s$  subsystems, and controls the subsystems separately based only on the local model and the information of the corresponding subsystem (see Fig. 3.2). By dividing the original system into subsystems and by designing decentralized controllers, the full information of the whole system is also separated into parts. The information interactions between subsystems are cut off, which results in a

non-classical information pattern:

$$I_i(t) = \{(\mathbf{u}_i, \mathbf{y}_i) | \mathbf{u}_i : [t_0, t] \rightarrow \mathbb{R}^{n_{u,i}} : \tau \mapsto \mathbf{u}_i(\tau), \mathbf{y}_i : [t_0, t] \rightarrow \mathbb{R}^{n_{y,i}} : \tau \mapsto \mathbf{y}_i(\tau)\}$$

$$\mathbf{u}_i(t) = f_i(I_i(t)) \quad \text{for each subsystem } i = 1, 2, \dots, s, \quad (3.3)$$

where  $\mathbf{u}_i$  and  $\mathbf{y}_i$  are the input vector and the output vector of subsystem  $i$ , and  $I_i(t)$  is the local information of subsystem  $i$ . So here, the vector of the control inputs to subsystem  $i$ ,  $\mathbf{u}_i(t)$ , depends only on its local information  $I_i(t)$ .

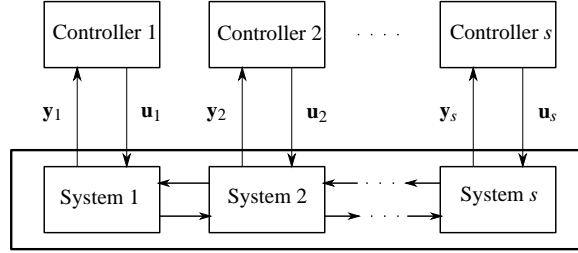


Figure 3.2: Decentralized control structure

Decentralized control can be opposed to centralized control. The computational complexity of the centralized MPC controller of a large urban traffic network can be reduced efficiently by dividing the network into small subnetworks, and controlling the local subnetwork MPC controllers separately in a decentralized structure. The traffic flow interactions between subnetworks are cut off (or disconnected), and will be considered constant and known by each subnetwork in advance. Because the estimates of the input traffic flows from other subnetworks may be far from the real values, the local MPC controllers may not be able to find the real optimal solutions for the subnetworks. Moreover, since the subnetworks are completely disconnected, the overall performance of the whole network will be deteriorated.

### 3.1.3 Distributed control structure

Similar to the decentralized control, distributed control also uses independent local controllers for different subsystems. Different from decentralized control, the local controllers exchange information and coordinate between each other. Therefore, each local controller will make its own decisions based on both information from the subsystem itself and the information obtained from other subsystems (see Fig. 3.3). The information pattern is non-classical and is expressed as

$$I_i(t) = \{(\mathbf{u}_j, \mathbf{y}_j) | \mathbf{u}_j : [t_0, t] \rightarrow \mathbb{R}^{n_{u,j}} : \tau \mapsto \mathbf{u}_j(\tau), \mathbf{y}_j : [t_0, t] \rightarrow \mathbb{R}^{n_{y,j}} : \tau \mapsto \mathbf{y}_j(\tau), j \in \mathcal{N}_i \cup \{i\}\}$$

$$\mathbf{u}_i(t) = f_i(I_i(t)) \quad \text{for each subsystem } i = 1, 2, \dots, s, \quad (3.4)$$

where  $I_i(t)$  is the local information of subsystem  $i$ ,  $\mathcal{N}_i$  is the set of the neighboring subsystems of subsystem  $i$  ("neighbor" can be pre-defined by the designer, which can be the adjacent subnetworks, or even including the subnetworks that are not directly connecting to

subnetwork  $i$ ), and  $\mathbf{u}_i(t)$  is the vector of the control inputs to subsystem  $i$ , which depends on its local information and the information from the neighboring subsystems of subsystem  $i$ .

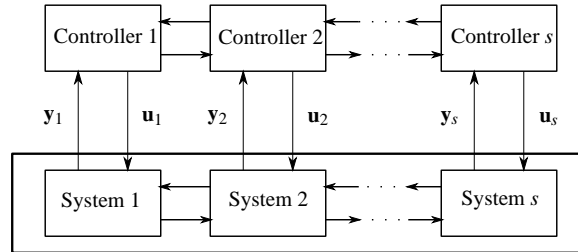


Figure 3.3: Distributed control structure

Distributed MPC offers a compromise between the decentralized MPC and the centralized MPC. It keeps the advantage of decentralized MPC as regards computational efficiency, and at the same time it also considers the status of other subsystems and tries to approach the overall performance of the centralized MPC. The more complete information the local MPC controllers have, the better overall performance of the whole traffic network will be achieved. However, if the amount of information that the local MPC controllers take in to consideration of increases, the computational complexity will become very high, and can then be comparable with the centralized MPC.

### 3.1.4 Hierarchical control structure

Hierarchical control (Fig. 3.4) is another control structure that tries to find a compromise between decentralized control and centralized control. Instead of giving all the control authority to local controllers, the hierarchical control structure divides the control problem into multiple control problems at multiple levels. On different levels, the controllers mainly aiming at solving specified different tasks. Generally speaking, the upper-level controllers will coordinate (or supervise) the subsystems from a global point of view. The lower-level controllers make decisions by themselves taking the advices from the upper-level controllers into consideration. Therefore, the information pattern of hierarchical control can contain two formats: on the coordination (or supervision) level, it can be a classical information pattern as the centralized controller; on the local level, it can be a non-classical information pattern either as in the decentralized controller or as in the distributed controller. Hierarchical control (see Fig. 3.4) allocates the control tasks to different control levels, on which control problems of different spatial/temporal scales are dealt with. This results in small-sized control problems to be solved at each control level.

According to the tasks allocated to the different control levels, the hierarchical control structure can be classified into two types:

- Bottom-up:  
If the local control decisions are made mainly by the local controllers themselves, and the upper-level controller is only responsible for letting the subnetworks communicate and coordinate with each other, then the hierarchical control structure is in a bottom-up format. The main control tasks are done by the local controllers, while the

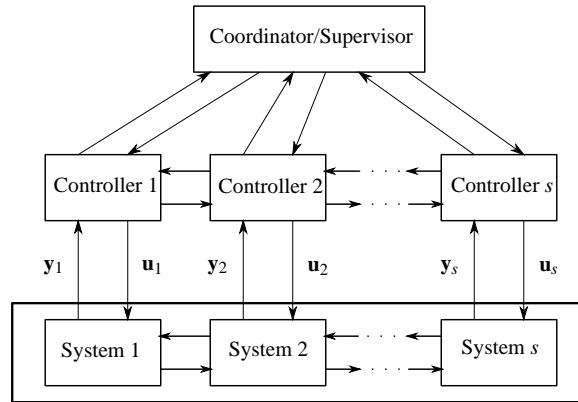


Figure 3.4: Hierarchical control structure

coordinator (or supervisor) will balance the resources, allocate the benefits among the subnetworks, etc.

- Top-down:

If the upper-level controller takes the main control task of the whole system like a centralized controller, it will make decisions for the whole network and give the commands to the local controllers, while the tasks of the local controllers are to decide how to carry out the commands obtained from the upper-level controller, then the hierarchical control structure is in a top-down format. Different from a centralized control problem, a top-down hierarchical control problem solves a more general control problem on the higher level, e.g. a control problem based on a more aggregate subnetwork input/output model, and then assigns the control results as the references for the lower level controllers. In this case, the supervisor from the upper level does not only coordinate the subnetworks, but also generates the global optimal solution for them, while the local controllers are just followers who will carry out the decisions.

Therefore, comparing all the control structures above, the distributed control structure and the hierarchical control structure can reduce the computational complexity of controlling a large-scale traffic network as a whole, and meanwhile also can achieve a trade-off between the overall control performance of the entire network and the local control performances of subnetworks. This multi-subnetwork control structure makes a modular design for the controller of each subnetwork possible, which enable the the network-wide controller to be expanded or reduced by adding or cutting subnetworks easily. As a result, both the flexibility and the scalability of the network-wide controller increase for the sake of the modular design for the multi-subnetwork control structure. Furthermore, the reliability, the sustainability, and the robustness of the traffic control system are also increased. Each local controller can, in the worst case, work independently. Thus, the control system will not break down, even if there are failures in other local controllers or even at the coordinator (or supervisor). However, for the top-down hierarchical structure, if the supervisor breaks down, the local controllers cannot work properly, but back-up local (i.e. decentralized or

distributed) control strategies can be switched on in the time of system failure. Moreover, a local controller can be maintained separately without influencing the others because of the independence. As a result, an MPC-based urban traffic control system is suitable to be implemented under a hierarchical or a distributed structure, as Fig. 3.5 shows.

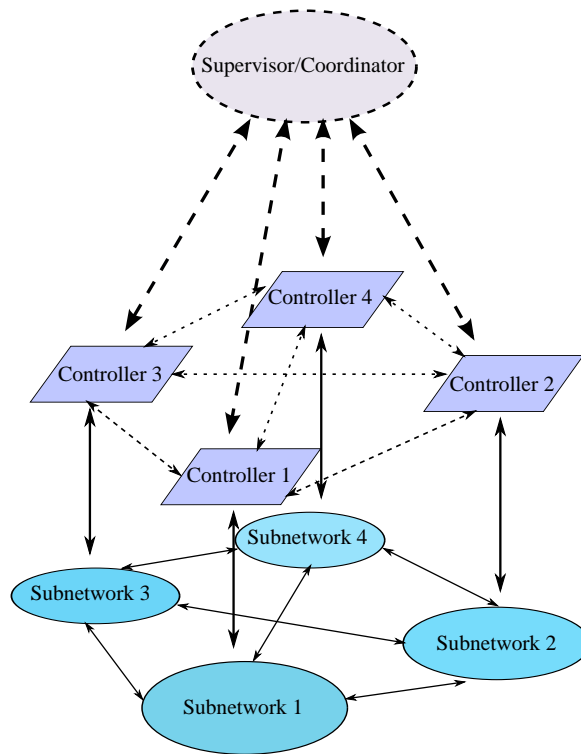


Figure 3.5: Distributed control structure for road networks

No matter which coordination scheme is selected for the traffic subnetworks, an advanced subnetwork controller is the foremost thing to be considered, so as to provide enough communication of information required by the coordination. Model Predictive Control (MPC) is chosen to be the control strategy for the subnetworks, because it can not only use the current traffic data, but also the predicted traffic information in the future.

## 3.2 General problem formulation

Model Predictive Control (MPC) is applied to control urban traffic networks. Any traffic model that can predict the future could be selected as the prediction model of MPC controllers. In the thesis, discrete-time macroscopic traffic flow models are used as prediction models of MPC controllers in the following expressions. The simulation time interval (i.e., the sampling time interval) of the discrete-time model is denoted by  $T_s$ . In order to control

the urban traffic network, a common control time interval is defined as  $T_c$ :

$$T_c = N \cdot T_s \quad (3.5)$$

with  $N$  an integer. Define  $k_s$  and  $k_c$  as step counters of the simulation time interval and the control time interval respectively. Note that these counters satisfy the following relation:

$$k_c = \left\lfloor \frac{k_s}{N} \right\rfloor, \quad (3.6)$$

where  $\lfloor x \rfloor$  with  $x \in \mathbb{R}$  denotes the largest integer smaller than or equal to  $x$ . In the remainder of this section, whenever  $k_s$  and  $k_c$  appear in the same equation, their relation is assumed to be given by (3.6).

The prediction models of the MPC controllers should be able to predict the future traffic states used for evaluating the objective function based on the information of current states, predicted demands, and future control inputs. They are generally described as

$$n(k_s + 1) = f(n(k_s), g(k_c), d(k_s)), \quad (3.7)$$

where  $n(k_s)$  is the traffic state needed for the objective function;  $d(k_s)$  is the traffic demand;  $g(k_c)$  is the future control input (e.g., the green time splits).

In the thesis, only the green time split is considered in the control measure. But, it is straightforward to add cycle time and offset as the control measures in the future. One of the advantages of MPC controllers is they can easily optimize and coordinate different control measures at the same time. Thus, the overall optimization problem of the whole urban traffic network for the MPC controller can be expressed as

$$\begin{aligned} \min_{\mathbf{g}(k_c)} J &= J_\theta(\hat{\mathbf{n}}(k_s), \mathbf{g}(k_c)) \\ \text{s.t. } \hat{\mathbf{n}}(k_s) &= \mathbf{f}(n(k_s), \mathbf{g}(k_c), \hat{\mathbf{d}}(k_s)); \\ \Phi(\mathbf{g}(k_c)) &= 0; \\ \mathbf{g}_{\min} &\leq \mathbf{g}(k_c) \leq \mathbf{g}_{\max}, \end{aligned} \quad (3.8)$$

where  $n(k_s)$  is the real or measured state of the network at time  $t = k_s \cdot T_s$ .

When the prediction horizon is  $N_p$ , then the predicted future traffic states are predicted at simulation time step  $k_s$  as

$$\hat{\mathbf{n}}(k_s) = [\hat{n}^T(k_s + 1|k_s) \hat{n}^T(k_s + 2|k_s) \cdots \hat{n}^T(k_s + NN_p|k_s)]^T,$$

based on the predicted traffic demands at simulation time step  $k_s$

$$\hat{\mathbf{d}}(k_s) = [\hat{d}^T(k_s|k_s) \hat{d}^T(k_s + 1|k_s) \cdots \hat{d}^T(k_s + NN_p - 1|k_s)]^T,$$

and the future traffic control inputs at control step  $k_c$

$$\mathbf{g}(k_c) = [g^T(k_c|k_c) g^T(k_c + 1|k_c) \cdots g^T(k_c + N_p - 1|k_c)]^T.$$

In (3.8), there are equality constraints and inequality constraints for the control inputs, where the inequality constraint puts an upper bound and a lower bound to the control inputs.

Different control objective functions can be selected for the MPC controller aiming at solving different traffic management problem. Objective  $J_\theta$  represents the objective function for performance  $\theta$ . Possible performance indices are Total Time Spent (TTS), Total Delay Time (TDT), Total Emissions (TE), etc.

**Remark 3.1** The objective function for the Total Time Spent (TTS) of the urban traffic network is

$$J_{\text{TTS}} = \sum_{(u,d) \in L} \sum_{k_s = Nk_c + 1}^{N(k_c + N_p)} T_s \cdot \hat{n}_{u,d}(k_s), \quad (3.9)$$

where  $\hat{n}_{u,d}(k_s)$  stands for the predicted number of vehicles on link  $(u, d)$  at simulation time step  $k_s$ .  $\square$

**Remark 3.2** The objective function for the Total Emissions (TE) of the urban traffic network is

$$J_{\text{TE}} = \sum_{(u,d) \in L} \sum_{k_s = Nk_c + 1}^{N(k_c + N_p)} \hat{E}_{u,d}(k_s), \quad (3.10)$$

where  $\hat{E}_{u,d}(k_s)$  stands for the predicted total vehicle emissions on link  $(u, d)$  during simulation time interval  $[k_s T_s, (k_s + 1) T_s]$ .  $\square$

The objective function of traffic subnetworks can be derived according to the above remarks. For the objective function of a large-scale traffic network with multiple subnetworks, we assume that the overall objective function equals the sum of all the subnetwork objective functions, as

$$J = \sum_{i \in \mathcal{S}} J_i, \quad (3.11)$$

which holds for all kinds of control performances, e.g. TTS and TE.

In order to reduce the on-line computational complexity, the large urban road network can be divided into several small subnetworks, and accordingly the overall optimization problem in (3.8) can be rewritten into sub-problems for each of the subnetworks. The set of the subnetworks is defined as  $\mathcal{S}$ . For the optimization problem (3.8), there are no couplings between subnetworks (i.e., all can be decomposed, including the objective function, the cycle time constraint, and the upper bounds and lower bounds of the green times), except the coupling terms between the models of the subnetworks, i.e. the traffic flows interactions among subnetworks. Therefore, the overall optimization problem can be decomposed directly into subnetwork optimization problems, if the interaction traffic flow constraints are ignored. The subnetwork optimization problems of subnetwork  $i$  and subnetwork  $j$  can be formulated as follows:

- Subnetwork  $i$ :

$$\begin{aligned} \min_{\mathbf{g}_i(k_c)} J_i &= \min_{\mathbf{g}_i(k_c)} J_{\theta,i}(\hat{\mathbf{n}}_i(k_s), \mathbf{g}_i(k_c)) \\ s.t. \quad \hat{\mathbf{n}}_i(k_s) &= \mathbf{f}(n_i(k_s), \mathbf{g}_i(k_c), \hat{\mathbf{d}}_i(k_s), \mathbf{z}_{j_1,i}(k_s), \mathbf{z}_{j_2,i}(k_s), \dots, \mathbf{z}_{j_{n_i},i}(k_s)); \\ \Phi_i(\mathbf{g}_i(k_c)) &= 0; \\ \mathbf{g}_{i,\min} &\leq \mathbf{g}_i(k_c) \leq \mathbf{g}_{i,\max}; \\ \mathbf{y}_{i,j}(k_s) &= \mathbf{f}_{i,\text{out}}(\hat{\mathbf{n}}_i(k_s), \mathbf{g}_i(k_c), \hat{\mathbf{d}}_i(k_s)), \quad \text{for all } j \in \mathcal{N}_i \end{aligned} \quad (3.12)$$

- Subnetwork  $j$ :

$$\begin{aligned}
\min_{\mathbf{g}_j(k_c)} J_j &= \min_{\mathbf{g}_j(k_c)} J_{\theta,j}(\hat{\mathbf{n}}_j(k_s), \mathbf{g}_j(k_c)) \\
s.t. \quad \hat{\mathbf{n}}_j(k_s) &= \mathbf{f}(n_j(k_s), \mathbf{g}_j(k_c), \hat{\mathbf{d}}_j(k_s), \mathbf{z}_{i_1,j}(k_s), \mathbf{z}_{i_2,j}(k_s), \dots, \mathbf{z}_{i_{n_j},j}(k_s)); \\
\Phi_j(\mathbf{g}_j(k_c)) &= 0; \\
\mathbf{g}_{j,\min} &\leq \mathbf{g}_j(k_c) \leq \mathbf{g}_{j,\max}; \\
\mathbf{y}_{j,i}(k_s) &= \mathbf{f}_{j,\text{out}}(\hat{\mathbf{n}}_j(k_s), \mathbf{g}_j(k_c), \hat{\mathbf{d}}_j(k_s)), \quad \text{for all } i \in \mathcal{N}_j
\end{aligned} \tag{3.13}$$

where the sets of the neighbor subnetworks of Subnetwork  $i$  and  $j$  are defined as  $\mathcal{N}_i = \{j_1, j_2, \dots, j_{n_i}\}$  and  $\mathcal{N}_j = \{i_1, i_2, \dots, i_{n_j}\}$ ,  $\mathbf{z}_{i,j}(k_s)$  stands for the vector of the input traffic flows running from subnetwork  $i$  into subnetwork  $j$ , and  $\mathbf{y}_{i,j}(k_s)$  represents for the vector of the output traffic flows running out of subnetwork  $i$  and then into subnetwork  $j$  (see Fig. 3.6). The output traffic flow  $\mathbf{y}_{i,j}(k_s)$  depends on the output function  $\mathbf{f}_{\text{out}}$  with respect to the traffic states, the traffic signal inputs, and the traffic demands of subnetwork  $i$ .

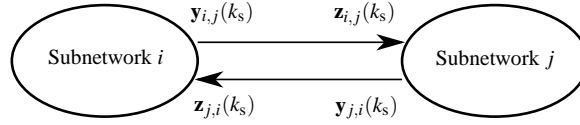


Figure 3.6: The interactions between subnetwork  $i$  and subnetwork  $j$

As in (3.12) and (3.13), the model of Subnetwork  $i$  also has to be updated based on the input traffic flow information provided by the neighbor subnetworks, i.e.  $\mathbf{z}_{j_1,i}(k_s), \mathbf{z}_{j_2,i}(k_s), \dots$ . It is also the same for the model of Subnetwork  $j$ . The interactions between subnetworks need to be guaranteed by extra interaction constraints. The interaction constraints between subnetworks cannot be added explicitly to the control problems of the subnetworks, but they can be adjusted and guaranteed by the upper-level coordinator [99]. The interaction constraints satisfy an *interaction balance condition*, which makes sure that the vector of the input traffic flows running from subnetwork  $i$  into subnetwork  $j$ ,  $\mathbf{z}_{i,j}(k_s)$ , equals to the vector of the output traffic flows running out of subnetwork  $i$  and then into subnetwork  $j$ ,  $\mathbf{y}_{i,j}(k_s)$ ; and vice versa. The interaction constraints can be formulated as

$$\mathbf{z}_{i,j}(k_s) = \mathbf{y}_{i,j}(k_s) \text{ for all } i \in \mathcal{S} \text{ and for all } j \in \mathcal{N}_i, \tag{3.14}$$

just as Fig. 3.6 illustrated. These interaction constraints guarantee that the traffic flows running out of subnetwork  $i$  equals the traffic flows getting into subnetwork  $j$  at each simulation step during the whole prediction horizon.

Coordination algorithms are needed for the subnetwork controllers to guarantee the interaction balance condition. However, due to the finite termination of the convergence process of the coordination algorithms, the interaction constraints can be only approximately satisfied. We will come back to this in Section 3.3. However, supposed all the interaction constraints among subnetworks are satisfied, and

$$\min J_i = J_i(\hat{\mathbf{n}}_i^*(k_s), \mathbf{g}_i^*(k_c)) \tag{3.15}$$



for all subnetworks, then in view of (3.11) the objective function of the whole traffic network converges as

$$\min J = \sum_{i \in \mathcal{S}} \min J_i = \sum_{i \in \mathcal{S}} J_i(\hat{\mathbf{n}}_i^*(k_s), \mathbf{g}_i^*(k_c)), \quad (3.16)$$

and the optimal green times for the overall network become

$$\mathbf{g}^*(k_c) = [\mathbf{g}_1^{*\text{T}}(k_c) \mathbf{g}_2^{*\text{T}}(k_c) \cdots \mathbf{g}_s^{*\text{T}}(k_c)]^{\text{T}} \quad (3.17)$$

where  $\mathcal{S} = \{1, 2, \dots, s\}$ .

### 3.3 Two coordination algorithms

In order to make sure that the interaction balance conditions among subnetworks are satisfied, the traffic subnetwork controllers need to communicate and coordinate with each other, and make an agreement. Algorithms for coordinating the subnetworks are necessary. The coordination algorithms under a distributed control structure can make a trade-off between the computational complexity of the controller and the overall control performance of the whole network. Of course, the coordination among subnetworks also introduces extra computations, but the computation effort for coordinating the subnetworks can be much less than the computational complexity of a centralized MPC for the entire traffic network, which in practice tends to increase exponentially with expansion of the network scale.

The dual optimization method (or the augmented Lagrangian method) [10, 20, 24] is a promising way to solve the coordination problem of subnetworks. First, we consider the overall optimization problem of the network that can be expressed as

$$\begin{aligned} \min_{\mathbf{g}(k_c)} J &= J_\theta(\hat{\mathbf{n}}(k_s), \mathbf{g}(k_c)) \\ \text{s.t. } \hat{\mathbf{n}}(k_s) &= \mathbf{f}(n(k_s), \mathbf{g}(k_c), \hat{\mathbf{d}}(k_s)); \\ \Phi(\mathbf{g}(k_c)) &= 0; \\ \mathbf{g}_{\min} &\leq \mathbf{g}(k_c) \leq \mathbf{g}_{\max}, \end{aligned} \quad (3.18)$$

with additional output functions for subnetworks, as

$$\mathbf{y}_{i,j}(k_s) = \mathbf{f}_{i,\text{out}}(\hat{\mathbf{n}}_i(k_s), \mathbf{g}_i(k_c), \hat{\mathbf{d}}_i(k_s)), \text{ for all } i \in \mathcal{S}, j \in \mathcal{N}_i, \quad (3.19)$$

where the set  $\mathcal{N}_i$  contains the indices of all the neighbor subnetworks of subnetwork  $i$ .

Then, by introducing in Lagrangian multiplier variable  $\omega$ , all the interaction constraints ( $\mathbf{z}_{j,i}(k_s) = \mathbf{y}_{j,i}(k_s)$ ) can be considered, the Lagrangian equation of the overall optimization problem with a relaxed interaction constraint term can be written as

$$L = J_\theta(\hat{\mathbf{n}}(k_s), \mathbf{g}(k_c)) + \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N}_i} \omega_{j,i}^{\text{T}}(k_s) (\mathbf{z}_{j,i}(k_s) - \mathbf{y}_{j,i}(k_s)). \quad (3.20)$$

The optimization problem can be then written into its dual optimization problem as

$$\begin{aligned}
& \max_{\omega(k_s)} \left( \min_{\mathbf{g}(k_c)} L \right) \\
s.t. \quad & \hat{\mathbf{n}}(k_s) = \mathbf{f}(n(k_s), \mathbf{g}(k_c), \hat{\mathbf{d}}(k_s)); \\
& \Phi(\mathbf{g}(k_c)) = \mathbf{0}; \\
& \mathbf{g}_{\min} \leq \mathbf{g}(k_c) \leq \mathbf{g}_{\max}; \\
& \mathbf{y}_{i,j}(k_s) = \mathbf{f}_{i,\text{out}}(\hat{\mathbf{n}}_i(k_s), \mathbf{g}_i(k_c), \hat{\mathbf{d}}_i(k_s)) \quad \text{for all } i \in \mathcal{S}, j \in \mathcal{N}_i. \quad (3.21)
\end{aligned}$$

By introducing in the Lagrangian multipliers, the interaction constraints are relaxed, and the overall dual problem is divisible into separate sub-problems if the Lagrangian multipliers are fixed. The Lagrangian multipliers are the interaction operators, which will help to coordinate the subnetworks, i.e. the subnetwork controllers will be punished if the interaction constraints are violated. Once the interaction operators are fixed, the Lagrangian term of (3.20) can be decomposed as

$$L = \sum_{i \in \mathcal{S}} L_i, \quad (3.22)$$

where

$$L_i = J_{\theta,i}(\hat{\mathbf{n}}_i(k_s), \mathbf{g}_i(k_c)) + \sum_{j \in \mathcal{N}_i} \omega_{j,i}^T \mathbf{z}_{j,i}(k_s) - \sum_{j \in \mathcal{N}_i} \omega_{i,j}^T \mathbf{y}_{i,j}(k_s). \quad (3.23)$$

Therefore, the overall optimization problem of (3.21) can be solved on two levels: on the lower level, subnetwork optimization problems can be solved independently with respect to only local variables when the interaction operators are fixed, while on the upper level, a global optimization problem will be solved to coordinate the interaction balance among subnetworks by adjusting the interaction operators. On the upper level, coordination algorithms are needed to make sure the interaction balance constraint,  $z_{j,i} = y_{j,i}$ , can be satisfied or approximated, even though the subnetwork optimization problems are solved separately. The lower and upper level optimization problems will be adjusted and then solved iteratively, until the interaction balance constraints are satisfied, or the finite termination condition is satisfied.

Within this multi-level control structure, there are two important approaches [95] to coordinate the subnetworks so as to approximate the interaction balance conditions:

- Interaction balance principle
- Interaction prediction principle

In the next subsections, these methods will be explained in more detail.

### 3.3.1 Interaction balance principle

In the interaction balance principle, the interactions among subnetworks are completely disconnected, and the input interaction variables  $\mathbf{z}_i(k_s)$  are considered as a new variable, which will be optimized by each of the subnetwork controllers. Therefore, the optimization

problem of the lower level becomes

$$\begin{aligned} & \min_{\mathbf{g}_i(k_c), \mathbf{z}_i(k_s)} L_i(\hat{\mathbf{n}}_i(k_s), \mathbf{g}_i(k_c), \mathbf{z}_i(k_s), \boldsymbol{\omega}(k_s)) \\ \text{s.t. } & \hat{\mathbf{n}}_i(k_s) = \mathbf{f}(n_i(k_s), \mathbf{g}_i(k_c), \hat{\mathbf{d}}_i(k_s), \mathbf{z}_{j_1,i}(k_s), \mathbf{z}_{j_2,i}(k_s), \dots, \mathbf{z}_{j_{n_i},i}(k_s)); \\ & \Phi_i(\mathbf{g}_i(k_c)) = 0; \\ & \mathbf{g}_{i,\min} \leq \mathbf{g}_i(k_c) \leq \mathbf{g}_{i,\max}; \end{aligned} \quad (3.24)$$

$$\mathbf{y}_{i,j}(k_s) = \mathbf{f}_{i,\text{out}}(\hat{\mathbf{n}}_i(k_s), \mathbf{g}_i(k_c), \hat{\mathbf{d}}_i(k_s)) \quad \text{for all } j \in \mathcal{N}_i \quad (3.25)$$

for subnetwork controller  $i$ , where  $\mathbf{z}_i(k_s)$  is the vector of input interaction variables for all the interactions with the neighbors of subnetwork  $i$  (for  $\mathcal{N}_i = \{j_1, j_2, \dots, j_{n_i}\}$ )

$$\mathbf{z}_i(k_s) = [\mathbf{z}_{j_1,i}^T(k_s) \mathbf{z}_{j_2,i}^T(k_s) \dots \mathbf{z}_{j_{n_i},i}^T(k_s)]^T, \quad (3.26)$$

and vector  $\boldsymbol{\omega}(k_s)$  contains the interaction operators of the whole network (subnetwork set  $\mathcal{S} = \{1, 2, \dots, s\}$ )

$$\begin{aligned} \boldsymbol{\omega}_i(k_s) &= [\boldsymbol{\omega}_{1,i}^T(k_s) \boldsymbol{\omega}_{2,i}^T(k_s) \dots \boldsymbol{\omega}_{n_i,i}^T(k_s)]^T \\ \boldsymbol{\omega}(k_s) &= [\boldsymbol{\omega}_1^T(k_s) \boldsymbol{\omega}_2^T(k_s) \dots \boldsymbol{\omega}_s^T(k_s)]^T, \end{aligned} \quad (3.27)$$

which is obtained from the upper level coordinator, and is considered constant during the local optimizations. On the lower level, such local optimization problems are solved for each subnetwork.

On the upper level, the interaction operators are revised according to the differences between the desired traffic flow input  $\mathbf{z}_i(k_s)$  and the real traffic flow supply  $\mathbf{y}_i(k_s)$  from neighboring subnetworks, so as to punish the subnetwork controllers if the interaction constraints cannot be satisfied, or to award the subnetwork controllers if the interaction constraints are satisfied. The output interaction vector  $\mathbf{y}_i(k_s)$  is defined corresponding to the input interaction vector  $\mathbf{z}_i(k_s)$  as

$$\mathbf{y}_i(k_s) = [\mathbf{y}_{j_1,i}^T(k_s) \mathbf{y}_{j_2,i}^T(k_s) \dots \mathbf{y}_{j_{n_i},i}^T(k_s)]^T. \quad (3.28)$$

The global optimization problem on the upper level becomes

$$\max_{\boldsymbol{\omega}(k_s)} (\varphi(\boldsymbol{\omega}(k_s))) = \max_{\boldsymbol{\omega}(k_s)} \left( \min_{\mathbf{g}(k_c)} L(\hat{\mathbf{n}}(k_s), \mathbf{g}(k_c), \mathbf{z}(k_s), \boldsymbol{\omega}(k_s)) \right). \quad (3.29)$$

The lower level optimization and the upper level optimization are calculated iteratively, until the interaction balance constraints are satisfied, or the finite termination condition is reached. In the signal,  $k$  is defined as the counter for the iterations, for instance,  $\mathbf{z}^k(k_s)$  and  $\mathbf{y}^k(k_s)$  are the input interaction variable and the output interaction variable at iteration  $k$ .

Since variables,  $\hat{\mathbf{n}}(k_s), \mathbf{g}(k_c), \mathbf{z}(k_s)$ , are already fixed on lower level subnetwork controllers, the gradient direction of the objective function of the upper level optimization can be calculated as

$$\nabla \varphi(\boldsymbol{\omega}(k_s)) = \begin{bmatrix} \mathbf{z}_1^k(k_s) - \mathbf{y}_1^k(k_s) \\ \mathbf{z}_2^k(k_s) - \mathbf{y}_2^k(k_s) \\ \vdots \\ \mathbf{z}_s^k(k_s) - \mathbf{y}_s^k(k_s) \end{bmatrix} = \mathbf{e}^k(k_s). \quad (3.30)$$

Therefore, the interaction operator  $\omega(k_s)$  can be updated for iteration  $k + 1$  towards the gradient direction as

$$\omega^{k+1}(k_s) = \omega^k(k_s) + \rho^k \mathbf{e}^k(k_s), \quad (3.31)$$

where  $\rho^k \in \mathbb{R}^+$  is a positive scalar that represents the search length, which can be a fixed value for all the iterations, or can vary and be optimized for each iteration. The iteration will end, when the error disappears or becomes very small, e.g. when the norm of the error satisfies the condition  $\|\mathbf{e}^k(k_s)\| < \varepsilon$  for  $\varepsilon > 0$ . The coordination structure of the interaction balance principle is shown in Fig. 3.7.

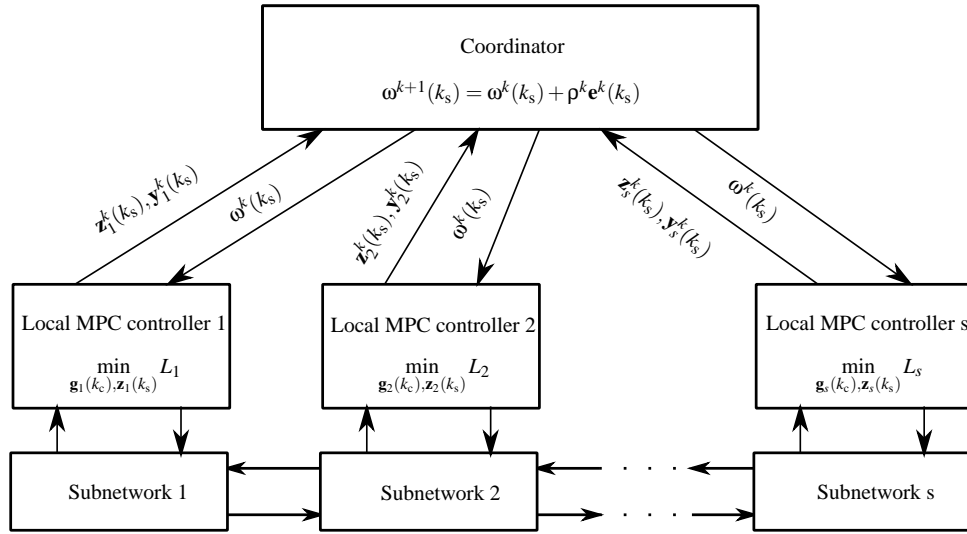


Figure 3.7: The coordination structure for the interaction balance principle

### 3.3.2 Interaction prediction principle

Different from the interaction balance principle, the interactions among subnetworks are not disconnected in interaction prediction principle, but are estimated by the coordinator based on the information from the neighboring subnetworks. In particular, the input traffic flows of subnetwork  $i$ , i.e.  $\mathbf{z}_i(k_s)$ , is not considered as a variable that needs to be optimized by the subnetwork controller  $i$ , but as a value that can be estimated based on the information provided by the neighboring subnetworks. Therefore, the optimization problem of the lower level becomes

$$\min_{\mathbf{g}_i(k_c)} L_i(\hat{\mathbf{n}}_i(k_s), \mathbf{g}_i(k_c), \mathbf{z}_i(k_s), \omega(k_s))$$

$$s.t. \quad \hat{\mathbf{n}}_i(k_s) = \mathbf{f}(n_i(k_s), \mathbf{g}_i(k_c), \hat{\mathbf{d}}_i(k_s), \mathbf{z}_{j_1,i}(k_s), \mathbf{z}_{j_2,i}(k_s), \dots, \mathbf{z}_{j_{n_i},i}(k_s));$$

$$\Phi_i(\mathbf{g}_i(k_c)) = 0;$$

$$\mathbf{g}_{i,\min} \leq \mathbf{g}_i(k_c) \leq \mathbf{g}_{i,\max}; \quad (3.32)$$

$$\mathbf{y}_{i,j}(k_s) = \mathbf{f}_{i,\text{out}}(\hat{\mathbf{n}}_i(k_s), \mathbf{g}_i(k_c), \hat{\mathbf{d}}_i(k_s)) \quad \text{for all } j \in \mathcal{N}_i \quad (3.33)$$

where both the interaction operator  $\omega$  and the vector of the traffic flows going into subnetwork  $i$ ,  $\mathbf{z}_i$ , are all estimated by the supervisor (the upper level controller), and are transferred to the lower level controller. When solving the lower level subnetwork optimization problems,  $\omega$  and  $\mathbf{z}_i$  are both considered constant.

On the upper level, the estimated traffic flow input at iteration  $k + 1$ , i.e.  $\mathbf{z}^{k+1}$ , can be updated directly by the real traffic flow output from neighboring subnetworks at current iteration  $k$  as

$$\mathbf{z}_i^{k+1}(k_s) = \mathbf{y}_i^k(k_s) \text{ for all } i \in \mathcal{S}. \quad (3.34)$$

In the same way as the interaction balance principle, the interaction operator can be updated by

$$\omega^{k+1}(k_s) = \omega^k(k_s) + \rho^k \mathbf{e}^k(k_s), \quad (3.35)$$

where  $\rho^k$  is also a positive scalar representing the update weight, which can be selected in a similar way as in the interaction balance principle. The errors between the desired traffic flow inputs and the real traffic flow supplies from neighboring subnetworks are calculated as

$$\mathbf{e}^k(k_s) = \begin{bmatrix} \mathbf{z}_1^k(k_s) - \mathbf{y}_1^k(k_s) \\ \mathbf{z}_2^k(k_s) - \mathbf{y}_2^k(k_s) \\ \vdots \\ \mathbf{z}_s^k(k_s) - \mathbf{y}_s^k(k_s) \end{bmatrix}.$$

According to (3.34), the error can also be written as

$$\mathbf{e}^k(k_s) = \begin{bmatrix} \mathbf{y}_1^{k-1}(k_s) - \mathbf{y}_1^k(k_s) \\ \mathbf{y}_2^{k-1}(k_s) - \mathbf{y}_2^k(k_s) \\ \vdots \\ \mathbf{y}_s^{k-1}(k_s) - \mathbf{y}_s^k(k_s) \end{bmatrix},$$

which means that the interaction operator  $\omega$  is updated according to the differences between the old output traffic flows  $\mathbf{y}^{k-1}$  and the newly obtained input traffic flows  $\mathbf{y}^k$ , so as to penalize the subnetwork controllers if the new obtained output traffic flows differ a lot from the output traffic flows from the previous iteration, or to end the iteration if the absolute difference between them is smaller than a given threshold.

The condition to end the coordination is the same as the interaction balance principle, i.e.  $\|\mathbf{e}^k(k_s)\| < \varepsilon$  for  $\varepsilon > 0$ . The coordination structure of the interaction prediction principle is shown in Fig. 3.8.

Both the interaction balance principle and interaction prediction principle are coordination algorithms for distributed control structures. They can be applied to coordinate MPC controllers of urban traffic subnetworks. Here are some remarks for these two coordination algorithms:

**Remark 3.3** In both the interaction balance principle and the interaction prediction principle, when the termination condition for the coordination is satisfied, if  $e^k = 0$ , then the optimal solution is achieved for the overall urban traffic network; if  $e^k \neq 0$ , then a sub-optimal solution will be obtained for the overall urban traffic network.  $\square$

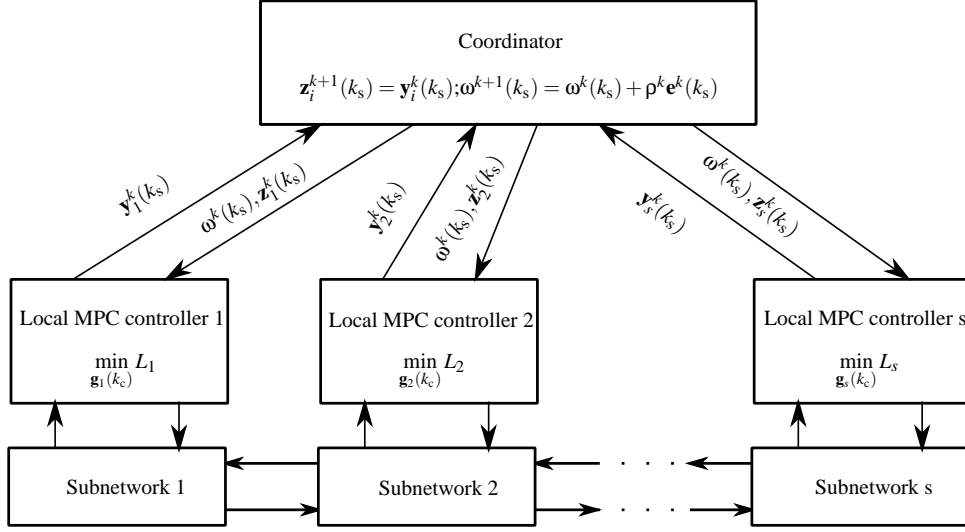


Figure 3.8: The coordination structure for interaction prediction principle

**Remark 3.4** The interaction balance principle endures more computational burden compared to the interaction prediction principle, since extra optimization variables ( $\mathbf{z}$ ) are introduced when completely disconnecting the interactions among subnetworks [95]. However, in this context, the interaction balance principle also gains more freedom for the subnetwork optimizations.  $\square$

**Remark 3.5** According to the coordination algorithms based on either the interaction balance principle or the interaction prediction principle, the more interaction constraints among the subnetworks, the more efforts are needed for coordination. Therefore, one principle of decomposing the urban traffic network is to make sure the number of interaction constraints among subnetworks is as small as possible.  $\square$

### 3.4 Summary

A well-defined control structure is necessary for controlling a complex, large-scale, urban traffic network, because of the existing layout structure of urban traffic networks, the dynamics of the urban traffic systems, the multiple control objectives, and the high computational complexity. By defining urban traffic control systems under specific control structures, the above problems of a complex large-scale network can be addressed more concretely.

The control structure for a large-scale network can be roughly classified into four types: centralized control structure, decentralized control structure, distributed control structure, and hierarchical control structure. The centralized control can achieve the best overall control performance of the whole network, but meanwhile suffers from high computational complexity (especially for the centralized MPC controller). The decentralized control divides the large urban traffic network into smaller subnetworks, and designs subnetwork

controllers for each subnetwork in a disconnected way from the rest of the subnetwork controllers. The computational complexity of subnetwork controllers is reduced significantly by dividing a large-scale network into smaller subnetworks, but the overall control performance of the whole network will inevitably deteriorate. To keep the simplicity of the subnetwork controllers and increase the overall control performance of the whole network, distributed control and hierarchical control are introduced. Distributed control increases the overall control performance by sharing and exchanging information among subnetwork controllers. In addition, hierarchical control improves the overall control performance by adding a coordinator or supervisor at a higher control level to guide the subnetwork controllers to agree on a global solution. Distributed control and hierarchical control are able to provide a trade-off between the centralized controller and the decentralized controller. Therefore, a complex large-scale urban traffic network can be better controlled under distributed control structures or hierarchical control structures [18, 39, 48, 70, 97].

After a large urban traffic network is divided into subnetworks, algorithms are needed to coordinate the subnetwork controllers so that the overall control performance of the original traffic network will be achieved. Two possible coordination principles based on the dual optimization method are discussed in this chapter. Extra interaction operators are introduced to guide the subnetwork MPC controllers to coordinate with each other iteratively, and the coordination will finish if all the input and the output traffic flow interaction constraints are satisfied within a pre-defined tolerance.

No matter which coordination scheme is selected for the traffic subnetworks, an advanced subnetwork controller is the foremost object to be considered, so as to achieve a good overall network control performance. Model Predictive Control (MPC) is chosen to be the control strategy for the subnetworks, because it can provide enough communication information, including not only the current traffic states, but also the predicted traffic states in the future. However, due to the on-line computational burden of MPC, the subnetwork controller may become infeasible in practice. An efficient MPC controller for the urban traffic subnetworks is always necessary. Therefore, in the rest of this thesis, we will investigate the efficiency and accuracy of urban traffic control models, and address the real-time feasibility problem of the MPC controllers for urban traffic subnetworks.





## Chapter 4

# Macroscopic Spatiotemporal Discrete Urban Traffic model

Traffic models that can predict future traffic states are the basis of model-based urban traffic predictive controllers. In this chapter, we are going to present macroscopic spatiotemporal discrete urban traffic control models, and to discuss about the features of the models with respect to different spatial and temporal sampling intervals<sup>1</sup>.

### 4.1 Introduction

Traffic models can be mainly classified into three categories based on the modeling details: microscopic models, macroscopic models, and mesoscopic models. Microscopic models are detailed traffic models that describe the dynamics of each individual vehicle, like car-following models. In contrast, macroscopic models are much rougher models focusing only on the dynamics of traffic flows, i.e. the average behavior of groups of vehicles instead of individual vehicles. Mesoscopic models combine both the properties of the microscopic models and the macroscopic models. A first-order macroscopic model was proposed by Lighthill and Whitham [79] to describe the dynamic of traffic flows, and it was extended into second-order macroscopic models [107]. But, this model was criticized for not being able to reproduce enough descriptive accuracy for modeling the phenomena of real traffic in [34]. In general, macroscopic models are approximations of traffic dynamics, and they ignore some details of individual vehicles and make a lot of simplifications, so macroscopic traffic models are in general not as accurate as the models with higher level-of-detail. However, this statement does not always hold in practice. On some occasions, macroscopic modeling approaches may provide better results than modeling approaches with a higher level-of-detail [63]. In addition, macroscopic models open a way for efficiently running the models using digital computers, and thus they are applied in traffic applications that are characterized by high computational requirements.

For different traffic applications, we need to select suitable traffic models with proper modeling accuracy and limited computation burden. Both microscopic models and macro-

---

<sup>1</sup>The content of this chapter was published in [80, 82, 87].

scopic models offer various levels of modeling power. Usually, the more detailed the traffic dynamics is modeled, the more complex the model will be, and the heavier computational burden the model will have. Therefore, when selecting a traffic model in practice, a criterion needs to be followed. The criterion [102] is that the model should have sufficient descriptive power to reproduce all important phenomena for the intended application, and at the same time the execution speed of a simulation should be fast enough for this particular application. That is to say we need to find a trade-off between the descriptive accuracy of the model and the computational complexity. This is the most important trade-off of the traffic models that we are focusing on in the thesis. The degree of non-linearity of the models is not really relevant with the analytical complexity of the models.

In urban areas, the traffic flows are influenced a lot by the traffic signals. Therefore, the store-and-forward model [52] was proposed to describe the stop-and-go traffic flow dynamics controlled by the traffic lights for urban roads. The store-and-forward model, later used for control by [40], is a simple model with a low computational complexity, but it only applies for saturated traffic, i.e. when the vehicle queues resulting from the red phase cannot be dissolved completely at the end of the following green phase. The model proposed by [8] and extended by [42] can describe vehicle queues and the time delay for vehicles reaching the queues in a link, and is able to describe different scenarios. Cell Transmission Model [33] and Link Transmission Model [128] are both models based on kinematic wave theory by Lighthill and Whitham [79], and Richards [109]. These two models are also spatiotemporally discrete traffic models. The model proposed by [66] has a lower modeling power, but cannot describe scenarios other than saturated traffic either. The model of [58, 80, 81, 119, 120] is capable of simulating the evolution of traffic dynamics (including vehicle queues) in all traffic scenarios (unsaturated, saturated, and over-saturated traffic conditions) by updating the discrete-time model in small simulation steps. To reduce the computational complexity of this model, [82, 83] proposed a model with a longer sampling time interval based on the previous model, but has intersection cycle times that can differ from intersection to intersection. The model is much faster than the previous model, with only a limited loss in modeling accuracy.

Actually, all the macroscopic urban traffic models mentioned above are spatiotemporally discrete models, which are spatially sampled into road segments and temporally sampled with a sampling time interval. For urban areas, the roads are comparatively short and divided by intersections with traffic lights, and thus an urban road is usually taken as a road segment. The sampling time interval can vary for different urban traffic models. A trade-off also needs to be made when selecting the sampling time interval for the discrete urban traffic model. Normally, a higher sampling frequency results in a more accurate model, but also gives rise to more computations because of having to update the model more frequently. When the sampling time interval becomes too large, the discrete model cannot represent the continuous traffic flow behavior anymore. Therefore, an additional criterion (Courant-Friedrichs-Lewy condition for urban traffic models) needs to be satisfied when sampling urban traffic models into spatiotemporally discrete models, so as to keep the descriptive ability of these models.

In this chapter, a spatiotemporally discrete urban traffic model with a variable sampling time interval is proposed for model-based predictive control, which allows to balance modeling accuracy and computational complexity. The discrete urban traffic model is derived by sampling the first-order continuous traffic flow model spatially and temporally. A CFL

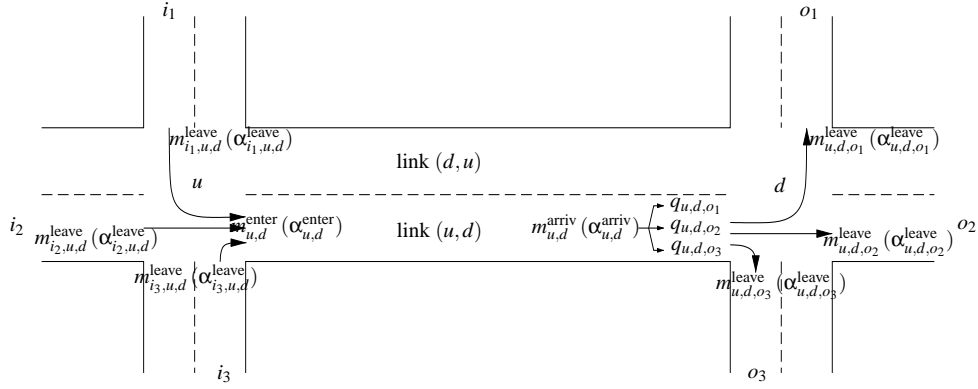


Figure 4.1: A link connecting two traffic-signal-controlled intersections

condition is deduced for the spatiotemporally discrete urban traffic model to make sure the descriptive ability of the model can be still guaranteed. Experiments are designed and evaluated to verify whether the models have sufficient descriptive power to reproduce the important phenomena for traffic control, and whether the computation speeds of models are fast.

## 4.2 Notations

In order to describe the model, we define  $J$  as the set of nodes (intersections), and  $L$  as the set of links (roads) in the urban traffic network. Link  $(u,d)$  is marked by its upstream node  $u$  ( $u \in J$ ) and downstream node  $d$  ( $d \in J$ ). The sets of the upstream nodes of input links and downstream nodes of output links for link  $(u,d)$  are  $I_{u,d} \subset J$  and  $O_{u,d} \subset J$  (e.g., for the situation of Fig. 4.1 we have  $I_{u,d} = \{i_1, i_2, i_3\}$  and  $O_{u,d} = \{o_1, o_2, o_3\}$ ).

The variable notations (see also Fig. 4.1) used in the models are listed as follows:

### General symbols

$I_{u,d}$	: set of upstream nodes of input links of link $(u,d)$ ,
$O_{u,d}$	: set of downstream nodes of output links of link $(u,d)$ ,
$T_c$	: control time interval,
$c_u$	: cycle time for Intersection $u$ ,
$c_d$	: cycle time for Intersection $d$ ,
$v_{u,d}^{\text{free}}$ (km/h)	: free-flow vehicle speed in link $(u,d)$ ,
$C_{u,d}$ (veh)	: capacity of link $(u,d)$ expressed in number of vehicles,
$N_{u,d}^{\text{lane}}$	: number of lanes in link $(u,d)$ ,
$\Delta c_{u,d}$ (s)	: offset between node $u$ and node $d$ , which represents the offset time between the cycles of the upstream and the downstream intersections at the beginning of every control time step,
$l_{\text{veh}}$ (m)	: average vehicle length.

**Spatiotemporally discrete model symbols**

$T_d$	: simulation time interval for Intersection $d$ ,
$k_d$	: simulation step counter for Intersection $d$ ,
$n_{u,d}(k_d)$ (veh)	: number of vehicles in link $(u, d)$ at step $k_d$ ,
$q_{u,d}(k_d)$ (veh)	: queue length (expressed as the number of vehicles) at step $k_d$ in link $(u, d)$ , $q_{u,d,o}$ is the queue length of the sub-stream turning to link $(d, o)$ ,
$S_{u,d}(k_d)$ (veh)	: available storage space of link $(u, d)$ at step $k_d$ expressed in number of vehicles,
$\alpha_{u,d}^{\text{leave}}(k_d)$ (veh/h)	: average flow rate leaving link $(u, d)$ at step $k_d$ , $\alpha_{u,d,o}^{\text{leave}}(k_d)$ is the leaving average flow rate of the sub-stream going towards link $(d, o)$ ,
$\alpha_{u,d}^{\text{arriv}}(k_d)$ (veh/h)	: average flow rate arriving at the tail of the queue in link $(u, d)$ at step $k_d$ , $\alpha_{u,d,o}^{\text{arriv}}(k_d)$ is the arriving average flow rate of the sub-stream going towards link $(d, o)$ ,
$\alpha_{u,d}^{\text{enter}}(k_d)$ (veh/h)	: average flow rate entering link $(u, d)$ at step $k_d$ ,
$\beta_{u,d,o}(k_d)$	: fraction of the traffic in link $(u, d)$ anticipating to turn to link $(d, o)$ at step $k_d$ ,
$\mu_{u,d,o}$ (veh/h)	: saturation flow rate leaving link $(u, d)$ turning to link $(d, o)$ ,
$g_{u,d,o}(k_d)$ (s)	: green time length during time interval $[k_d T_d, (k_d + 1) T_d]$ for the traffic stream towards link $(d, o)$ in link $(u, d)$ .

**BLX model symbols**

$T_s$	: simulation time interval for BLX model,
$k$	: simulation step counter for the BLX model,
$n_{u,d}(k)$ (veh)	: number of vehicles in link $(u, d)$ at step $k$ ,
$q_{u,d}(k)$ (veh)	: queue length (expressed as the number of vehicles) at step $k$ in link $(u, d)$ , $q_{u,d,o}$ is the queue length of the sub-stream turning to link $(d, o)$ ,
$m_{u,d,o}^{\text{leave}}(k)$ (veh)	: number of vehicles leaving link $(u, d)$ and turning to link $(d, o)$ at time interval $[k T_s, (k + 1) T_s]$ ,
$m_{u,d}^{\text{arriv}}(k)$ (veh)	: number of vehicles arriving <i>at the (tail of the) queue</i> in link $(u, d)$ at time interval $[k T_s, (k + 1) T_s]$ , $m_{u,d,o}^{\text{arriv}}(k)$ is the number of arriving cars in the sub-stream going towards link $(d, o)$ ,
$m_{u,d}^{\text{enter}}(k)$ (veh)	: number of vehicles entering link $(u, d)$ at time interval $[k T_s, (k + 1) T_s]$ ,
$S_{u,d}(k)$ (veh)	: available storage space of link $(u, d)$ at step $k$ expressed in number of vehicles,
$\beta_{u,d,o}(k)$	: fraction of the traffic in link $(u, d)$ anticipating to turn to link $(d, o)$ at step $k$ ,
$\mu_{u,d,o}$ (veh/h)	: saturation flow rate leaving link $(u, d)$ turning to link $(d, o)$ ,
$b_{u,d,o}(k)$ (s)	: green time length during time interval $[k T_s, (k + 1) T_s]$ for the traffic stream towards link $(d, o)$ in link $(u, d)$ , which can be considered as a boolean value indicating whether the traffic signal at intersection $d$ for the traffic stream in link $(u, d)$ turning to link $(d, o)$ is green (1) or red (0) if $T_s = 1$ s.

**S model symbols**

$c_d$	: simulation time interval for Intersection $d$ ,
$k_d$	: simulation step counter for Intersection $d$ ,
$n_{u,d}(k_d)$ (veh)	: number of vehicles in link $(u,d)$ at step $k_d$ ,
$q_{u,d}(k_d)$ (veh)	: queue length (expressed as the number of vehicles) at step $k_d$ in link $(u,d)$ , $q_{u,d,o}$ is the queue length of the sub-stream turning to link $(d,o)$ ,
$\alpha_{u,d}^{\text{leave}}(k_d)$ (veh/h)	: average flow rate leaving link $(u,d)$ at step $k_d$ , $\alpha_{u,d,o}^{\text{leave}}(k_d)$ is the leaving average flow rate of the sub-stream going towards link $(d,o)$ ,
$\alpha_{u,d}^{\text{arriv}}(k_d)$ (veh/h)	: average flow rate arriving at the tail of the queue in link $(u,d)$ at step $k_d$ , $\alpha_{u,d,o}^{\text{arriv}}(k_d)$ is the arriving average flow rate of the sub-stream going towards link $(d,o)$ ,
$\alpha_{u,d}^{\text{enter}}(k_d)$ (veh/h)	: average flow rate entering link $(u,d)$ at step $k_d$ ,
$\beta_{u,d,o}(k_d)$	: fraction of the traffic in link $(u,d)$ anticipating to turn to link $(d,o)$ at step $k_d$ ,
$\mu_{u,d,o}$ (veh/h)	: saturation flow rate leaving link $(u,d)$ turning to link $(d,o)$ ,
$g_{u,d,o}(k_d)$ (s)	: green time length during time interval $[k_d c_d, (k_d + 1)c_d]$ for the traffic stream towards link $(d,o)$ in link $(u,d)$ .

**4.3 Discrete-time delay**

In this chapter, the urban traffic models are discrete-time models with a time delay, during which a vehicle travels from the beginning of the road until it reaches the queues waiting in the road. In [2], a method is presented to sample a continuous-time system with a time delay into a discrete-time system. Given this method, the discrete-time delay, which the vehicles take to reach the end of the queues in a link, will be obtained. Let a linear continuous time-invariant system with time delay  $\gamma \in \mathbb{R}^+$  be described by<sup>2</sup>

$$\dot{\tilde{\mathbf{X}}}(t) = \mathbf{A}\tilde{\mathbf{X}}(t) + \mathbf{B}\tilde{\mathbf{U}}(t - \gamma) . \quad (4.1)$$

Let us now sample this system using a sampling period  $T$ . Define

$$\delta = \text{floor} \left\{ \frac{\tau}{T} \right\}, \quad \gamma = \text{rem} \{ \tau, T \}, \quad (4.2)$$

where  $\text{floor}\{x\}$  refers to the largest integer smaller than or equal to  $x$ , and  $\text{rem}\{x,y\}$  is the remainder of the division of  $x$  by  $y$ . So  $\delta$  is an integer, and the time delay  $\tau$  can be expressed as

$$\tau = \delta \cdot T + \gamma \quad 0 \leq \gamma < T . \quad (4.3)$$

If the input of the system ( $\tilde{\mathbf{U}}(t)$ ) is assumed to be piece-wise constant during each sampling time interval, the sampled discrete-time system will be

$$\mathbf{X}(k+1) = \Phi\mathbf{X}(k) + \Gamma_0\mathbf{U}(k - \delta) + \Gamma_1\mathbf{U}(k - \delta - 1) , \quad (4.4)$$

<sup>2</sup>: represents a continuous variable.

where

$$\Gamma_0 = \int_0^{T-\gamma} e^{As} ds \mathbf{B} \quad (4.5)$$

$$\Gamma_1 = e^{A(T-\gamma)} \int_0^\gamma e^{As} ds \mathbf{B} . \quad (4.6)$$

Thus, the vehicles that enter into a link normally will run with free-flow speed for a certain time, and finally join the tail of the queues. This time period is a time delay that is needed before the vehicles join the queues waiting at the stop-line of the link. Then, the queue length in a link is updated by the number of vehicles leaving the link and the number of delayed vehicles entering the link. The differential equation describing the evolution of the queue length can be therefore written as

$$\dot{q}_{u,d,o}(t) = \tilde{\beta}_{u,d,o}(t) \tilde{\alpha}_{u,d}^{\text{enter}}(t - \tau) - \tilde{\alpha}_{u,d,o}^{\text{leave}}(t), \quad (4.7)$$

i.e. the changing rate of the queue length ( $\dot{q}_{u,d,o}(t)$ ) is equal to the difference between the input flow rate (delayed by  $\tau$  and then divided by multiplying the current turning rate) and the output flow rate. In (4.7), the traffic flow turning rate ( $\tilde{\beta}_{u,d,o}(t)$ ), and the traffic flow rate entering or leaving the queue ( $\tilde{\alpha}_{u,d}^{\text{enter}}(t)$  and  $\tilde{\alpha}_{u,d,o}^{\text{leave}}(t)$ ), are all piece-wise constant during the sampling time intervals. The traffic flow turning rate will be influenced by the traffic flows and traffic signals the drivers experienced upstream, the traffic signals in front, and the origin-destination of the drivers. Then, according to the addition principle of linear equations, (4.7) can be divided into two equations, as

$$\dot{q}_{u,d,o}^1(t) = -\tilde{\alpha}_{u,d,o}^{\text{leave}}(t) \quad (4.8)$$

$$\dot{q}_{u,d,o}^2(t) = \tilde{\beta}_{u,d,o}(t) \tilde{\alpha}_{u,d}^{\text{enter}}(t - \tau), \quad (4.9)$$

such that

$$\tilde{q}_{u,d,o}(t) = \tilde{q}_{u,d,o}^1(t) + \tilde{q}_{u,d,o}^2(t). \quad (4.10)$$

To sample differential equation (4.8) without a time delay into a discrete equation, we define  $A = 0$  and  $B = -1$ , then according to (4.4), (4.5), and (4.6), we have

$$q_{u,d,o}^1(k+1) = \Phi q_{u,d,o}^1(k) + \Gamma \alpha_{u,d,o}^{\text{leave}}(k) \quad (4.11)$$

where

$$\begin{aligned} \Phi &= e^{AT} = 1 \\ \Gamma &= \int_0^T e^{As} ds \mathbf{B} = -T \end{aligned} \quad (4.12)$$

Similarly, we can sample differential equation (4.9) with a time delay  $\tau$  into a discrete equation. Since, in Section 4.4 the time delay  $\tau$  will vary slowly with time  $t$ , then according to (4.2) and (4.3) we can approximately have

$$\delta(k) = \text{floor} \left\{ \frac{\tau(k)}{T} \right\}, \quad \gamma(k) = \text{rem} \{ \tau(k), T \}, \quad (4.13)$$

and

$$\tau(k) = \delta(k) \cdot T + \gamma(k) \quad 0 \leq \gamma(k) < T. \quad (4.14)$$

Next, we define  $A_\tau = 0$  and  $B_\tau = 1$ , and then according to (4.4), (4.5), and (4.6), (4.9) results in

$$\begin{aligned} q_{u,d,o}^2(k+1) = & \Phi_\tau q_{u,d,o}^2(k) + \beta_{u,d,o}(k) (\Gamma_0 \alpha_{u,d}^{\text{enter}}(k - \delta(k)) \\ & + \Gamma_1 \alpha_{u,d}^{\text{enter}}(k - \delta(k) - 1)), \end{aligned} \quad (4.15)$$

where

$$\begin{aligned} \Phi_\tau &= e^{A_\tau} = 1 \\ \Gamma_0 &= \int_0^{T-\gamma(k)} e^{A_\tau s} \mathbf{B}_\tau = T - \gamma(k) \\ \Gamma_1 &= e^{A_\tau(T-\gamma(k))} \int_0^{\gamma(k)} e^{A_\tau s} \mathbf{B}_\tau = \gamma(k) \end{aligned} \quad (4.16)$$

Therefore, by adding (4.11) and (4.15) together, we derive

$$\begin{aligned} q_{u,d,o}(k+1) = & q_{u,d,o}(k) - T \alpha_{u,d,o}^{\text{leave}}(k) \\ & + \beta_{u,d,o}(k) ((T - \gamma(k)) \alpha_{u,d}^{\text{enter}}(k - \delta(k)) \\ & + \gamma(k) \alpha_{u,d}^{\text{enter}}(k - \delta(k) - 1)), \end{aligned} \quad (4.17)$$

and the arriving average traffic flow at the tail of the queues

$$\alpha_{u,d}^{\text{arriv}}(k) = \frac{T - \gamma(k)}{T} \alpha_{u,d}^{\text{enter}}(k - \delta(k)) + \frac{\gamma(k)}{T} \alpha_{u,d}^{\text{enter}}(k - \delta(k) - 1). \quad (4.18)$$

## 4.4 Spatiotemporally discrete urban traffic model

In this section, we will derive a spatiotemporally discrete urban traffic model with a variant sampling time interval.

### 4.4.1 Traffic dynamics on a link

Suppose the sampling time interval for intersection  $d \in J$  and all the links that connect to intersection  $d$  is  $T_d$  and  $k_d$  is the corresponding time step counter. Due to the physical structure of urban networks, the original urban roads are directly taken as spatially sampled link segments. For the sake of simplicity, only controlled intersections are considered, but it can also extended to un-controlled intersections.

**Assumption 4.1** *The cycle time of intersection  $j (\in J)$  can be defined as*

$$c_j = M_j T_j, \quad (4.19)$$

where  $M_j$  and  $T_j$  are integers, and  $0 < T_j \leq c_j$ . Sampling time intervals and cycle times can be different for intersections.

**Assumption 4.2** *We assume parallel turning lanes exist in the traffic model. The vehicles getting into a link will run on the link freely without turning separations, until they reach the tail of the waiting vehicle queues. Once they reach the tail of the queues, they will be divided to join the queues of the turning direction they intend to go.*

Therefore, a spatiotemporally discrete urban traffic model can be derived as follows:

The number of the vehicles in link  $(u, d)$  is updated by the input and output average flow rate over sampling time interval  $T_d$  at every time step  $k_d$  by

$$n_{u,d}(k_d + 1) = n_{u,d}(k_d) + \left( \alpha_{u,d}^{\text{enter}}(k_d) - \alpha_{u,d}^{\text{leave}}(k_d) \right) \cdot T_d, \quad (4.20)$$

and consequently we can update the storage capacity as

$$S_{u,d}(k_d) = C_{u,d} - n_{u,d}(k_d). \quad (4.21)$$

The leaving average flow rate is the sum of the leaving flow rates turning to each output link:

$$\alpha_{u,d}^{\text{leave}}(k_d) = \sum_{o \in O_{u,d}} \alpha_{u,d,o}^{\text{leave}}(k_d). \quad (4.22)$$

The leaving average flow rate over  $T_d$  is determined by:

$$\alpha_{u,d,o}^{\text{leave}}(k_d) = \min \left( \mu_{u,d,o} \cdot g_{u,d,o}(k_d) / T_d, \right. \\ \left. q_{u,d,o}(k_d) / T_d + \alpha_{u,d,o}^{\text{arriv}}(k_d), \right. \\ \left. \frac{\mu_{u,d,o}}{\sum_{u' \in I_{d,o}} \mu_{u',d,o}} \cdot \frac{C_{d,o} - n_{d,o}(k_d)}{T_d} \right), \quad (4.23)$$

where  $\mu_{u,d,o}$  is the saturation flow rate that can leave link  $(u, d)$  turning to link  $(d, o)$  depending on the physical structure of link  $(u, d)$ . The leaving flow rate is the minimum value of three flow rate values, average saturated flow rate, average unsaturated flow rate, and average over-saturated flow rate, which are given respectively by the three formulas in (4.23). The first term calculates the average saturated flow rate, which depends on the saturation flow rate  $\mu_{u,d,o}$  and green time duration; the second term calculates the average unsaturated flow rate based on the vehicles waiting in and arriving the queues; the third term calculates the average over-saturated flow rate depending on the proportional storage capacity of the downstream link. In some cases, there exists an inconsistency between the downstream storage capacity and the traffic flow demands of upstream links, but this inconsistency can be checked and solved according to the approaches given in [30] and [58, Chapter 8].

The number of vehicles waiting in the queue turning to link  $(d, o)$  is updated as

$$q_{u,d,o}(k_d + 1) = q_{u,d,o}(k_d) + \left( \alpha_{u,d,o}^{\text{arriv}}(k_d) - \alpha_{u,d,o}^{\text{leave}}(k_d) \right) \cdot T_d. \quad (4.24)$$

Here we made an assumption that the vehicles getting into a link do not separate for their turning directions. They run on the link freely until they reach the tail of the waiting



vehicle queues. Then they will join the queues of the turning direction they intend to go. Thus, the number of waiting vehicles in link  $(u, d)$  is

$$q_{u,d}(k_d) = \sum_{o \in O_{u,d}} q_{u,d,o}(k_d) . \quad (4.25)$$

The flow rate entering link  $(u, d)$  will arrive at the end of the queues after a time delay

$$\tau(k_d) = \frac{(C_{u,d} - q_{u,d}(k_d)) \cdot l_{\text{veh}}}{N_{u,d}^{\text{lane}} \cdot v_{u,d}^{\text{free}} \cdot T_d} , \quad (4.26)$$

then with  $\delta(k_d)$  and  $\gamma(k_d)$  derived from formulas (4.2) and (4.3), according to (4.17) the delayed flow rate arriving at the end of queues is

$$\begin{aligned} \alpha_{u,d}^{\text{arriv}}(k_d) &= \frac{T_d - \gamma(k_d)}{T_d} \cdot \alpha_{u,d}^{\text{enter}}(k_d - \delta(k_d)) + \\ &\quad \frac{\gamma(k_d)}{T_d} \cdot \alpha_{u,d}^{\text{enter}}(k_d - \delta(k_d) - 1) . \end{aligned} \quad (4.27)$$

Before reaching the tail of the waiting queues in link  $(u, d)$ , the flow rate of arriving vehicles need be divided by multiplying it with the turning rates:

$$\alpha_{u,d,o}^{\text{arriv}}(k_d) = \beta_{u,d,o}(k_d) \cdot \alpha_{u,d}^{\text{arriv}}(k_d) . \quad (4.28)$$

The flow rate entering link  $(u, d)$  is made up from the flow rates from all the input links:

$$\alpha_{u,d}^{\text{enter}}(k_d) = \sum_{i \in I_{u,d}} \alpha_{i,u,d}^{\text{leave}}(k_d) . \quad (4.29)$$

If  $c_d \neq c_u$ , then  $\alpha_{i,u,d}^{\text{leave}}(k_d)$  cannot be directly obtained from upstream links. Thus, synchronization between the intersections need to be further addressed.

#### 4.4.2 Synchronization between two intersections

In (4.29), the flow rate entering link  $(u, d)$  is provided by the combination of the flow rates leaving the upstream links. Recall that we may have different sampling time intervals between upstream and downstream intersections ( $T_u \neq T_d$ ). Thus, the simulation time steps may be not equal to each other. Therefore, in order to synchronize the traffic flows in the links connecting to the upper and the downstream intersections, it is necessary to synchronize the leaving and entering flow rates. First of all, a least common multiple time interval has to be defined as

$$T_{\text{lcm}} = N_j \cdot c_j \quad \text{for all } j \in J, \quad (4.30)$$

with  $N_j$  an integer, as Fig. 4.2(a) shows.

Then, in each time interval  $T_{\text{lcm}}$ , we will recast the flow rates expressed in the timing of intersection  $u$  into the timing of intersection  $d$ . As illustrated in Fig. 4.2(b), first, we transform the discrete time leaving flow rates from the upstream links into continuous time using the zero-order hold strategy, as

$$\tilde{\alpha}_{i,u,d}^{\text{leave}}(t) = \alpha_{i,u,d}^{\text{leave}}(k_u), \quad k_u \cdot T_u \leq t < (k_u + 1) \cdot T_u, \quad (4.31)$$

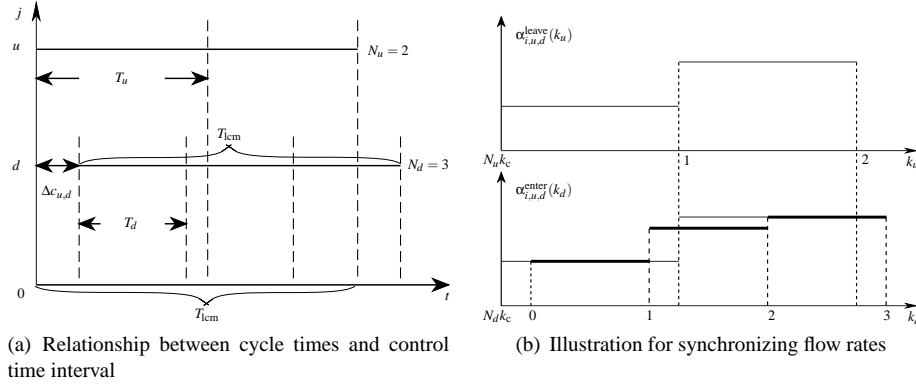


Figure 4.2: Synchronization of upstream and downstream intersections

and then sample them again to obtain the average flow rates in time step  $k_d$  so as to be able used by the downstream link

$$\alpha_{i,u,d}^{\text{enter}}(k_d) = \frac{1}{T_d} \int_{k_d \cdot T_d}^{(k_d+1) \cdot T_d + \Delta c_{u,d}} \alpha_{i,u,d}^{\text{leave}}(t) dt, \quad (4.32)$$

where  $\Delta c_{u,d}$  represents the offset time between the cycle times of the upstream and the downstream intersections at the beginning of a control time step. Then, the flow rate entering link  $(u, d)$  can be computed by

$$\alpha_{u,d}^{\text{enter}}(k_d) = \sum_{i \in I_{u,d}} \alpha_{i,u,d}^{\text{enter}}(k_d). \quad (4.33)$$

**Remark 4.1** The BLX model [80, 82] and the S model [82] proposed in previous research work can be qualitatively considered as special cases of this spatiotemporally discrete urban traffic model. But, they have different sampling time intervals (BLX:  $T_j = 1$  s, S:  $T_j = c_j$ ). In addition, the BLX model is updated by the input and output number of vehicles, while the S model is updated by the input and output average flow rates. Since the relationship between the average flow rate and the number of vehicles entering or leaving a link is

$$\alpha_{u,d}(k_d) = m_{u,d}(k_d)/T_d, \quad (4.34)$$

where  $m_{u,d}(k_d)$  represents the number of vehicles leaving or arriving link  $(u, d)$  at time step  $k_d$ , while  $\alpha_{u,d}(k_d)$  represents the corresponding average traffic flow rate during this time interval. Therefore, it is equivalent to update the discrete model with the vehicles entering or leaving a link in term of either the number of vehicles or the average flow rates.  $\square$

## 4.5 CFL condition for urban traffic models

The Courant-Friedrichs-Lewy condition (CFL condition) [31] is a necessary condition for convergence while solving certain partial differential equations (PDEs) (usually hyperbolic

PDEs) numerically. In general, it is not a sufficient condition for convergence. The CFL condition for one dimensional case can be expressed as

$$\frac{u\Delta t}{\Delta x} \leq C, \quad (4.35)$$

where  $u$  is the velocity of flow dynamics,  $\Delta t$  is the time step size,  $\Delta x$  is the spatial step size, and  $C$  is a constant scale parameter. Equation (4.35) makes sure that the time step must be less than a certain value, otherwise the simulation will produce wildly incorrect results.

We will derive the CFL condition for spatiotemporally discrete urban traffic models as follows:

The maximum number of vehicles that can leave link  $(u, d) \in L$  with a saturation flow rate (also called as link-intersection capacity) should not exceed the number of vehicles on this link, that is

$$\mu_{u,d}T_d \leq n_{u,d}(k_d) \leq C_{u,d}, \quad (4.36)$$

where the number of vehicles on link  $(u, d)$  is bounded by its storage capacity  $C_{u,d}$ , and the link-intersection capacity  $\mu_{u,d}$  is the sum of the saturation flow rates that leave link  $(u, d)$  turning into different directions:

$$\mu_{u,d} = \sum_{o \in O_{u,d}} \mu_{u,d,o}. \quad (4.37)$$

Then, by dividing the number of vehicles on link  $(u, d)$  into two parts, the number of vehicles in the queue ( $q_{u,d}(k_d)$ ) and the number of vehicles running freely on the link ( $f_{u,d}(k_d)$ ), as Fig. 4.3 shows, we have

$$T_d \leq \frac{n_{u,d}(k_d)}{\mu_{u,d}} = \frac{q_{u,d}(k_d) + f_{u,d}(k_d)}{\mu_{u,d}}. \quad (4.38)$$

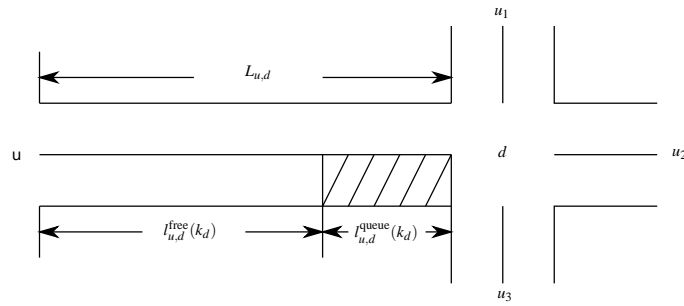


Figure 4.3: Illustration for the free-speed flow and the queues

According to traffic theory, if we define  $\alpha_{u,d}^{\text{free}}$  as the traffic flow rate for the vehicles freely running on link  $(u, d)$  before joining the tail of the vehicle queues, then the traffic flow running with free-flow speed on the link always has a lower flow rate than the link-intersection capacity, i.e.  $\alpha_{u,d}^{\text{free}}(k_d) \leq \mu_{u,d}$ , and the traffic flow rate for the vehicles moving

in queues is also lower than the link-intersection capacity, i.e.  $\alpha_{u,d}^{\text{queue}}(k_d) \leq \mu_{u,d}$ . Hence, (4.38) can be further written into

$$\begin{aligned} T_d &\leq \frac{q_{u,d}(k_d)}{\alpha_{u,d}^{\text{queue}}(k_d)} + \frac{f_{u,d}(k_d)}{\alpha_{u,d}^{\text{free}}(k_d)} \\ &= \frac{\rho_{u,d}^{\text{queue}}(k_d) l_{u,d}^{\text{queue}}(k_d)}{\alpha_{u,d}^{\text{queue}}(k_d)} + \frac{\rho_{u,d}^{\text{free}}(k_d) l_{u,d}^{\text{free}}(k_d)}{\alpha_{u,d}^{\text{free}}(k_d)} \\ &= \frac{l_{u,d}^{\text{queue}}(k_d)}{v_{u,d}^{\text{queue}}(k_d)} + \frac{l_{u,d}^{\text{free}}(k_d)}{v_{u,d}^{\text{free}}}, \end{aligned} \quad (4.39)$$

where  $\rho_{u,d}^{\text{queue}}(k_d)$  and  $\rho_{u,d}^{\text{free}}(k_d)$  are the density of the queue and the density of the free-running traffic flow on link  $(u,d)$  at time step  $k_d$  respectively. Furthermore, because the length of link  $(u,d)$  equals to the sum of the queue length and the free-running link length, i.e.  $l_{u,d}^{\text{queue}}(k_d) + l_{u,d}^{\text{free}}(k_d) = L_{u,d}$ , and  $v_{u,d}^{\text{queue}}(k_d) \leq v_{u,d}^{\text{free}}$ , we have

$$\frac{l_{u,d}^{\text{queue}}(k_d)}{v_{u,d}^{\text{queue}}(k_d)} + \frac{l_{u,d}^{\text{free}}(k_d)}{v_{u,d}^{\text{free}}} \leq \frac{L_{u,d}}{v_{u,d}^{\text{queue}}(k_d)}. \quad (4.40)$$

Since the average speed of the vehicles waiting in queues is bounded as  $0 \leq v_{u,d}^{\text{queue}}(k_d) \leq v_{u,d}^{\text{free}}$ , then (4.40) can be further written as

$$\frac{l_{u,d}^{\text{queue}}(k_d)}{v_{u,d}^{\text{queue}}(k_d)} + \frac{l_{u,d}^{\text{free}}(k_d)}{v_{u,d}^{\text{free}}} \leq \min\left(\frac{L_{u,d}}{v_{u,d}^{\text{queue}}(k_d)}\right) = \frac{L_{u,d}}{v_{u,d}^{\text{free}}}. \quad (4.41)$$

Hence, we derive a sufficient condition for the sampling time interval  $T_d$  of the model, as

$$T_d \leq \frac{L_{u,d}}{v_{u,d}^{\text{free}}}, \quad (4.42)$$

which is exactly a CFL condition. The condition can be interpreted intuitively as requiring that the distance  $v_{u,d}^{\text{free}} T_d$  traveled by a traffic flow in one time step should not exceed one spatial step  $\Delta x$ , or equivalently that the numerical traffic flow speed  $L_{u,d}/T_d$  should be at least as fast as the physical traffic flow speed  $v_{u,d}^{\text{free}}$ . In practice, a CFL condition can be used as a criterion for selecting proper sampling time intervals for the spatiotemporally discrete traffic models.

However, for urban intersections, the sampling time intervals of intersection  $d$  not only depend on the link  $(u,d)$ , but also on the rest of the links connecting to this intersection. We define  $U_d \subset J$  is the set of the possible upstream intersections of intersection  $d \in J$ . Therefore, to guarantee that the spatiotemporally discrete urban traffic model can correctly represent the urban traffic dynamics, the simulation time interval  $T_d$  (i.e. sampling time interval) needs to satisfy condition:

$$T_d \leq \min_{u' \in U_d} \left( \frac{L_{u',d}}{v_{u',d}^{\text{free}}} \right). \quad (4.43)$$

## 4.6 BLX model and S model

In this section we present the original model of M. van den Berg et al. [119] and S. Lin et al. [80] (indicated as the BLX model) as well as a new simplified model [82] (called the S model). According to Remark 4.1, the BLX model and the S model both can be qualitatively considered as special cases of the spatiotemporally discrete urban traffic model described in previous sections.

### 4.6.1 BLX model

In the BLX model a queue is modeled as follows. For the sake of simplicity, the assumption is made that at an intersection the cars going to the same destination move into the correct lane, so that they do not block the traffic flows going to other destinations. For each lane (or destination), a separate queue is constructed (with queue lengths denoted by  $q$ ). Furthermore, the simulation time step  $T_s$  is typically set to 1 s and cars arriving at the end of a queue in simulation period  $[kT_s, (k+1)T_s)$  are allowed to cross the intersection in that same period (provided that they have green, that there is enough space in the destination link, and that there are no other restrictions).

Consider link  $(u, d)$  (see Fig. 4.1). For each  $o \in O_{u,d}$  the number of cars leaving link  $(u, d)$  for destination  $o$  in the period  $[kT_s, (k+1)T_s)$  is given by

$$m_{u,d,o}^{\text{leave}}(k) = \begin{cases} 0 & \text{if } b_{u,d,o}(k) = 0 \\ \max(0, \min(\mu_{u,d,o}T_s, S_{d,o}(k), q_{u,d,o}(k) + m_{u,d,o}^{\text{arriv}}(k))) & \text{if } b_{u,d,o}(k) = 1. \end{cases} \quad (4.44)$$

The traffic arriving at the tail of the queue in link  $(u, d)$  is given by the traffic entering the link via the upstream intersection delayed by the time  $\tau(k) \cdot T_s + \gamma(k)$  needed to drive from the upstream intersection to the end of the queue in the link; to this extent  $m_{u,d}^{\text{arriv}}$  is updated as follows:

$$m_{u,d}^{\text{arriv}}(k) = \frac{T_s - \gamma(k)}{T_s} \cdot \sum_{(i,u) \in I_{u,d}} m_{i,u,d}^{\text{leave}}(k - \tau(k)) + \frac{\gamma(k)}{T_s} \cdot \sum_{(i,u) \in I_{u,d}} m_{i,u,d}^{\text{leave}}(k - \tau(k) - 1), \quad (4.45)$$

where

$$\tau(k) = \text{floor} \left\{ \frac{S_{u,d}(k) \cdot l_{\text{veh}}}{N_{u,d}^{\text{lane}} \cdot v_{u,d}^{\text{free}} \cdot T_s} \right\}, \quad \gamma(k) = \text{rem} \left\{ S_{u,d}(k) \cdot l_{\text{veh}}, N_{u,d}^{\text{lane}} \cdot v_{u,d}^{\text{free}} \cdot T_s \right\}. \quad (4.46)$$

The fraction of the arriving traffic in link  $(u, d)$  turning to  $o \in O_{u,d}$  is

$$m_{u,d,o}^{\text{arriv}}(k) = \beta_{u,d,o}(k) \cdot m_{u,d}^{\text{arriv}}(k). \quad (4.47)$$

The new queue lengths are given by the old queue lengths plus the arriving traffic minus the leaving traffic

$$q_{u,d,o}(k+1) = q_{u,d,o}(k) + m_{u,d,o}^{\text{arriv}}(k) - m_{u,d,o}^{\text{leave}}(k) \quad (4.48)$$

for each  $o \in O_{u,d}$ , and

$$q_{u,d}(k) = \sum_{o \in O_{u,d}} q_{u,d,o}(k) . \quad (4.49)$$

The new available storage stage depends on the number of cars that enter and leave the link in the period  $[kT_s, (k+1)T_s)$ :

$$S_{u,d}(k+1) = S_{u,d}(k) - \sum_{i \in I_{u,d}} m_{i,u,d}^{\text{leave}}(k) + \sum_{o \in O_{u,d}} m_{u,d,o}^{\text{leave}}(k) . \quad (4.50)$$

This BLX model is derived by extending the model of M. van den Berg et al. [119]. The difference between the BLX model and M. van den Berg's model is equation (4.45). In (4.45), the tail of the waiting queues in a link is fixed, and the number of arriving vehicles joining the queues is calculated by the number of vehicles entered the link a certain time delay before (i.e. historical data). On the contrary, in M. van den Berg's thesis, the calculation is made based on the time when vehicles enter the link, and predict the number of vehicles join the queue in future. Better results were obtained by the BLX model compared with M. van den Berg's model (see [80]).

However, we found that there is a problem with the BLX model that it consumes too much computation time, even though as a macroscopic model. It makes the MPC controller based on BLX model impossible to be used in practice. Therefore, we come up with the idea to find a more simple and fast model, which results in the S model. The S model has a larger sampling time interval, and updates by the average traffic flow rates over the time interval. It also allows different cycle times for intersections. The computing time was largely reduced by the S model compared with the BLX model. Therefore, it makes the MPC controllers possible to be implemented in urban traffic networks.

#### 4.6.2 Simplified model (S model)

In the simplified model, every intersection takes the cycle time as its simulation time interval. The cycle times for intersection  $u$  and  $d$ , which are denoted by  $c_u$  and  $c_d$  respectively, can be different from each other, as Fig. 4.2(a) illustrates. Moreover, the S model works with (average) flow rates rather than with number of cars for describing flows leaving or entering links.

Taking the cycle time  $c_d$  as the length of the simulation time interval for link  $(u,d)$  and  $k_d$  as the corresponding time step counter, the number of the vehicles in link  $(u,d)$  is updated according to the input and output average flow rate over  $c_d$  at every time step  $k_d$  by

$$n_{u,d}(k_d+1) = n_{u,d}(k_d) + \left( \alpha_{u,d}^{\text{enter}}(k_d) - \alpha_{u,d}^{\text{leave}}(k_d) \right) \cdot c_d . \quad (4.51)$$

The leaving average flow rate is the sum of the leaving flow rates turning to each output link:

$$\alpha_{u,d}^{\text{leave}}(k_d) = \sum_{o \in O_{u,d}} \alpha_{u,d,o}^{\text{leave}}(k_d) , \quad (4.52)$$

which keeps positive in reality ( $\alpha_{u,d}^{\text{leave}}(k_d) \geq 0$ ).

The leaving average flow rate over  $c_d$  is determined by the capacity of the intersection, the number of cars waiting and/or arriving, and the available space in the downstream link:

$$\alpha_{u,d,o}^{\text{leave}}(k_d) = \min \left( \begin{aligned} &\mu_{u,d,o} \cdot g_{u,d,o}(k_d) / c_d, \\ &q_{u,d,o}(k_d) / c_d + \alpha_{u,d,o}^{\text{arriv}}(k_d), \\ &\beta_{u,d,o}(C_{d,o} - n_{d,o}(k_d)) / c_d \end{aligned} \right). \quad (4.53)$$

The number of vehicles waiting in the queue turning to  $o$  is updated as

$$q_{u,d,o}(k_d + 1) = q_{u,d,o}(k_d) + \left( \alpha_{u,d,o}^{\text{arriv}}(k_d) - \alpha_{u,d,o}^{\text{leave}}(k_d) \right) \cdot c_d. \quad (4.54)$$

Then, the number of waiting vehicles in link  $(u, d)$  is

$$q_{u,d}(k_d) = \sum_{o \in O_{u,d}} q_{u,d,o}(k_d). \quad (4.55)$$

The flow rate entered link  $(u, d)$  will arrive at the tail of the queues after a time delay  $\tau(k_d) \cdot c_d + \gamma(k_d)$ , i.e.,

$$\alpha_{u,d}^{\text{arriv}}(k_d) = \frac{c_d - \gamma(k_d)}{c_d} \cdot \alpha_{u,d}^{\text{enter}}(k_d - \tau(k_d)) + \frac{\gamma(k_d)}{c_d} \cdot \alpha_{u,d}^{\text{enter}}(k_d - \tau(k_d) - 1), \quad (4.56)$$

$$\begin{aligned} \tau(k_d) &= \text{floor} \left\{ \frac{(C_{u,d} - q_{u,d}(k_d)) \cdot l_{\text{veh}}}{N_{u,d}^{\text{lane}} \cdot v_{u,d}^{\text{free}} \cdot c_d} \right\}, \\ \gamma(k_d) &= \text{rem} \left\{ (C_{u,d} - q_{u,d}(k_d)) \cdot l_{\text{veh}}, N_{u,d}^{\text{lane}} \cdot v_{u,d}^{\text{free}} \cdot c_d \right\}. \end{aligned} \quad (4.57)$$

Before reaching the tail of the waiting queues in link  $(u, d)$ , the flow rate of arriving vehicles need be divided by multiplying the turning rates:

$$\alpha_{u,d,o}^{\text{arriv}}(k_d) = \beta_{u,d,o}(k_d) \cdot \alpha_{u,d}^{\text{arriv}}(k_d). \quad (4.58)$$

The flow rate entering link  $(u, d)$  is made up from the flow rates from all the input links:

$$\alpha_{u,d}^{\text{enter}}(k_d) = \sum_{i \in I_{u,d}} \alpha_{i,u,d}^{\text{leave}}(k_d). \quad (4.59)$$

In this formula, we see that the flow rate entering link  $(u, d)$  is provided by the combination of the flow rates leaving the upstream links. Recall that we have different cycle times between the upstream and downstream intersections, so the simulation time steps are not the same. Some operations need to be carried out to synchronize the leaving and entering flow rates.

Thus,  $T_{\text{lcm}}$  is defined as the least common multiple of all the intersection cycle times in the traffic network. As Fig. 4.2(a) shows, we have

$$T_{\text{lcm}} = N_j \cdot c_j, \quad (4.60)$$

with  $N_j$  an integer, and equation  $N_u \cdot c_u = N_d \cdot c_d$  can be satisfied.

In order to control the urban traffic network, a common control time interval need to be specified for the network model, so that intersections can communicate with each other and be synchronous.

$$T_c = N \cdot T_{\text{lcm}} \quad (4.61)$$

with  $N$  an integer.

Now we show how the flow rates expressed in the timing of intersection  $u$  can be recast into the timing of intersection  $d$ . First, we smooth the leaving flow rates from the upstream links as

$$\alpha_{i,u,d}^{\text{leave,C}}(t) = \alpha_{i,u,d}^{\text{leave}}(k_u), \quad k_u \cdot c_u \leq t < (k_u + 1) \cdot c_u, \quad (4.62)$$

and then sample them again to obtain the average flow rates in time step  $k_d$  so as to be able used by the downstream link, as Fig. 4.2(b) shows:

$$\alpha_{i,u,d}^{\text{enter}}(k_d) = \frac{1}{c_d} \int_{k_d \cdot c_d + \Delta c_{u,d}}^{(k_d+1) \cdot c_d + \Delta c_{u,d}} \alpha_{i,u,d}^{\text{leave,C}}(t) dt. \quad (4.63)$$

Then, (4.59) can be computed instead as follows:

$$\alpha_{u,d}^{\text{enter}}(k_d) = \sum_{i \in I_{u,d}} \alpha_{i,u,d}^{\text{enter}}(k_d). \quad (4.64)$$

## 4.7 Model assessment

In this section, we will evaluate the effectiveness of this spatiotemporally discrete urban traffic model, and analyze its sensitivity from a control point of view. Experiments are designed to demonstrate how the Total Time Spent (TTS, frequently selected as traffic control performance criterion) will change by varying green time lengths of traffic signals. The evaluated urban road network is shown in Fig. 4.4. It is a simple urban road network with 3 intersections, and 8 origins. The origins, marked as ‘‘Ox’’, are the origin nodes where traffic flows enter the network. The evaluation performance criterion is the Total Time Spent (TTS), which is the accumulated time spent by all the vehicles in a region of the road network for the entire simulation time. If the region is the entire road network, then it is the TTS of the network; If the region is only a link, then it is the TTS of that particular link.

As Fig. 4.4 shows, the length of the roads in the network are 450 m and 900 m, and all the links have 3 lanes. The vehicle anticipating turning rates  $\beta$  are constant, i.e. 0.33 for left turn, through turn, and right turn respectively. The saturation flow rates  $\mu$  are 1800 veh/h, 1600 veh/h, and 1500 veh/h respectively for turning through, left, and right in each link. The average vehicle length  $l_{\text{veh}}$  is set to 7 m, and the free-flow speed  $v_{u,d}^{\text{free}}$  is 50 km/h. Then the storage capacities  $C$  are 193 veh for link (1,2) and link (2,1), and 386 veh for the rest of the links in the network. Fixed-time control is executed for each intersection, where the phases, the cycle times, and the green time lengths are all constant during each simulation.



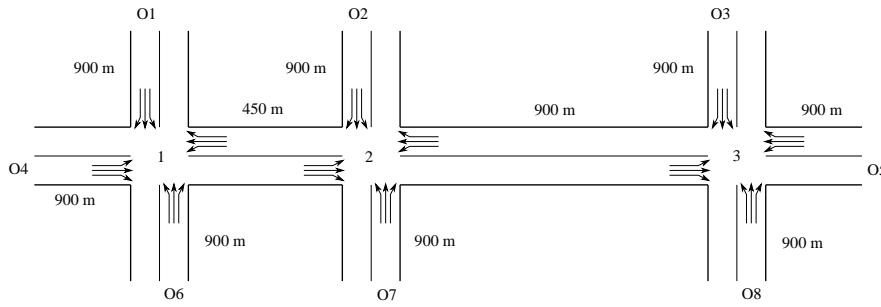


Figure 4.4: Layout of an urban road network

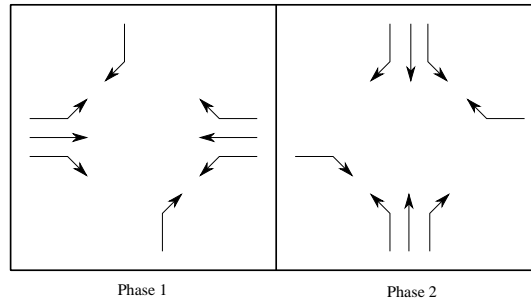


Figure 4.5: Intersection traffic signal phases

The phases and their order for all the intersections are given in Fig. 4.5. The green time lengths and cycle times are shown in Table 4.1. The symbol  $g_{j,i}$  stands for the green time length of the  $i$ th phase for intersection  $j$ . The offsets between intersections are specified as 0. The network input flow rates of the subnetwork are set to be equal to each other and constant in time (2000 veh/h). The simulation time duration is 30 min.

In order to evaluate how the evaluation performance (TTS) changes with the traffic signals, we allow the green time lengths of intersection 2 and 3 to change within a given time region,  $g_{2,1}, g_{3,1} \in \{15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75\}$ . The lower bound and the upper bound for a green time duration is 15 s and 75 s, and  $g_{2,2}$  and  $g_{3,2}$  change accordingly with  $g_{2,1}$  and  $g_{3,1}$ , due to the cycle time constraint of each intersection. The proposed spatiotemporally discrete traffic model is sampled by different sampling time intervals (simulation time intervals), i.e.  $T = 1$  s, 30 s, and 90 s respectively. Then, for each set-up of the traffic signals, all the sampled discrete traffic models are run for the same period of time

Table 4.1: Traffic signal fixed-time control setup

Intersection	Phase 1 (s)	Phase 2 (s)	Cycle time (s)
1	45	45	90
2	$g_{2,1}$	$90 - g_{2,1}$	90
3	$g_{3,1}$	$90 - g_{3,1}$	90

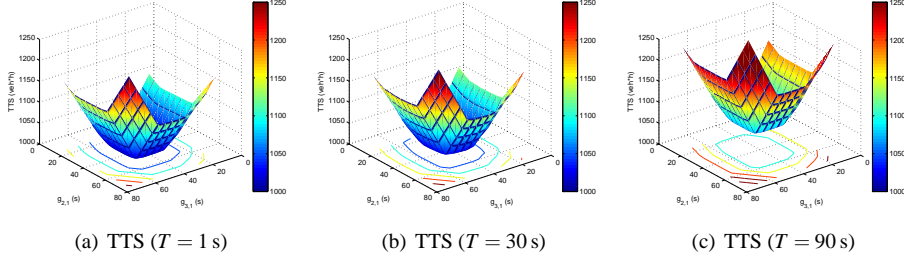


Figure 4.6: TTS of the network in Fig. 4.4 for discrete model with different sampling time intervals

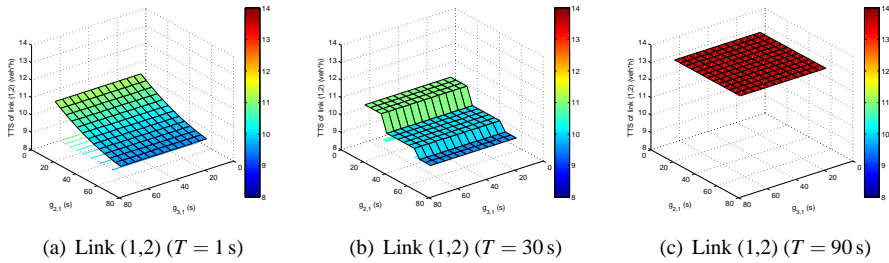


Figure 4.7: TTS of link (1,2) in Fig. 4.4 for discrete model with different sampling time intervals

(30 min). According to the CFL condition for urban traffic models in Section 4.5, the upper bounds of the sampling time intervals for every intersection are  $T_{1,\max} = T_{2,\max} = 32$  s and  $T_{3,\max} = 64$  s. Therefore, when  $T_1 = T_2 = T_3 = 30$ , the urban CFL conditions are satisfied in all the three intersections, as  $30 < T_{1,\max} = T_{2,\max} < T_{3,\max}$ ; when  $T_1 = T_2 = T_3 = 90$ , the urban CFL condition is violated. The comparison of the results are shown in Fig. 4.6 and Fig. 4.7 for the TTS of the entire network and for the TTS of link (1,2), in which the urban CFL condition is easier to be violated. If the boundary links are full, extra vehicles coming are stored, and are released if there is space available. These stored vehicles were also counted when calculating the TTS.

From Fig. 4.6 and Fig. 4.7, we can see that the spatiotemporally discrete traffic model can describe a more detailed variation of the TTS changing with the green time lengths, when the sampling time interval is small. For  $T = 1$  s and  $T = 30$  s, the shapes of the TTS curves are very similar to each other for both the entire network and the single link (1,2). Although the surface in Fig. 4.7(b) is not as smooth as the surface in Fig. 4.7(a), the scales of the values shown in both figures are the same. Generally speaking, the larger the sampling time interval is, the faster the model will run. For the discrete traffic models with sampling time intervals as 1 s and 30 s, the time needed to run the simulation are 5.6 s and 0.4 s respectively. Consequently, the discrete model with  $T = 30$  s is a better choice for urban traffic network control, because it can guarantee almost the same performance as the

discrete model with  $T = 1$  s, but requires less computing time.

For the discrete model with sampling time 90 s, the time needed to run the simulation is even less, 0.2 s. But, the sampling time becomes too large so as to violate the CFL condition. Thus, the model fail to describe the correct variation of the traffic phenomenon. In Fig. 4.6(c), the values of the TTS become much higher than that of the discrete models with  $T = 1$  s and  $T = 30$  s. In Fig. 4.7(c), the TTS curve becomes very flat, which cannot capture the variation of TTS values anymore. Therefore, even though the discrete model with  $T = 90$  s is very fast, but it does not have sufficient accuracy to be used as a control model. Consequently, in this case study, the spatiotemporally discrete urban traffic model with sampling time  $T = 30$  s is comparatively more suitable to be used as a prediction model for the urban traffic controllers, which gives a good trade-off between the modeling accuracy and the computational complexity.

If we reduce the length of link (1,2) and link (2,1) from 450 m to 150 m, then the CFL condition is even tighter for intersection 1 and 2, i.e. the upper bounds of the sampling time interval for intersection 1 and 2 become  $T_{1,\max} = T_{2,\max} = 10$  s. Thus, even though the sampling time interval is selected as  $T = 30$  s, the CFL condition is also violated. In such condition, the comparisons of the TTS of the entire network and of the TTS of link (1,2) are shown in Fig. 4.8 and Fig. 4.9.

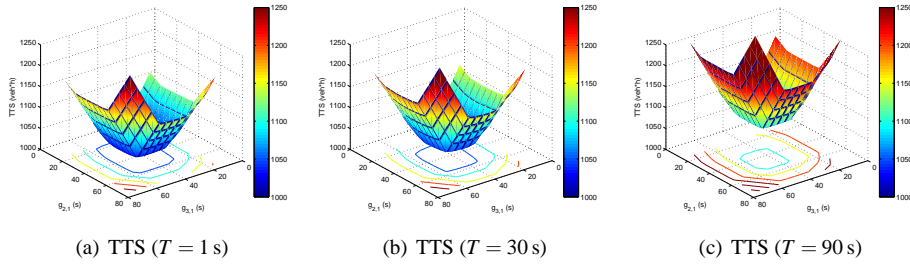


Figure 4.8: TTS of the network for the spatiotemporally discrete model with different sampling time intervals when the length of link (1,2) equals 150 m

As Fig. 4.8 and Fig. 4.9 illustrate, similar conclusions can be derived as above. In addition, since the CFL condition is violated when  $T = 30$  s, Fig. 4.9(b) also fails to follow the trend of the curve in Fig. 4.9(a). As the CFL condition suggests, the shorter the network links are, the higher the free-flow speed is, and the smaller a sampling time interval is needed to provide enough accuracy for the spatiotemporally discrete urban traffic model.

When green times are set as  $g_{2,1} = 45$  s and  $g_{3,1} = 45$  s, the evolution of the number of vehicles in link (2,1) is shown in Fig. 4.10 for different sampling time intervals. As the figure illustrates, for all situations, there exists spillback on link (2,1), where the number of vehicles in link (2,1) reaches its storage capacity, and will block the departures from upstream links. However, when  $T = 90$  s, the spillback cannot occur as fast as the spillback illustrated on the curves when  $T = 1$  s and  $T = 30$  s.

In urban networks, the turning movements depend on the decisions of individual drivers, which are with large uncertainty. Actually, in the models of this thesis, the turning move-

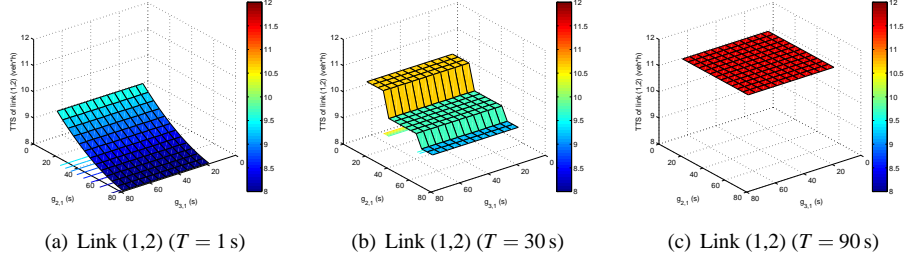


Figure 4.9: TTS of link (1,2) for the spatiotemporally discrete model with different sampling time intervals when the length of link (1,2) equals 150 m

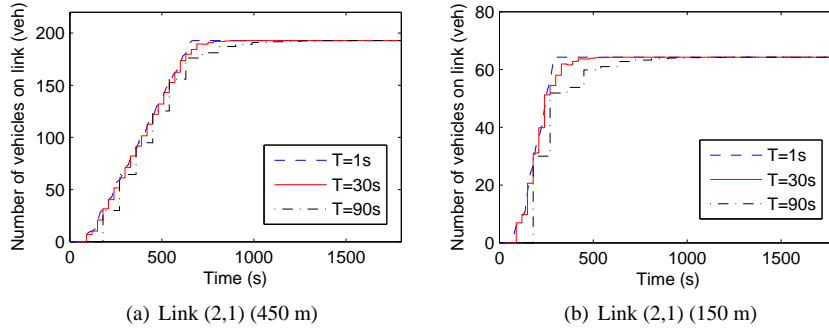


Figure 4.10: The evolution of the number of vehicles in link (2,1)

ments are related to the turning rates, which are parameters for the models. Due to the high complexity, the parameters are assumed to be constant for the models in this thesis. In fact, these parameters should not be static, but will vary with time. In reality, these parameters will change with time for different O-D allocations, different route guidance information, different effects of weather conditions, seasonal variations, events (like concerts or soccer games), etc. Therefore, for future work, we are going to investigate time-varying urban traffic control models that will adapt their parameters according to the dynamic traffic information of the network. This was also proposed as one of the recommendations for future research in Chapter 8. We will also add uncertainty to future simulations on testing the time-varying urban traffic models.

## 4.8 Summary

Traffic models that can predict future traffic states are the basis for establishing model-based traffic control strategies. Model-based control, especially Model Predictive Control (MPC), requires fast models due to the use of on-line optimization. Therefore, a suitable prediction

traffic model that is both accurate enough and fast enough, is needed.

Consequently, macroscopic urban traffic models are selected as prediction models. In this chapter, a macroscopic spatiotemporally discrete urban traffic model with a variable sampling time interval is proposed for model-based predictive control. Applying varying sampling time intervals allows to balance the modeling accuracy and the computational complexity of the discrete traffic models, and allows to search for the best trade-off for specific control requirements. A CFL condition is deduced for the spatiotemporally discrete urban traffic model to make sure the descriptive ability of the model can be still guaranteed, when the sampling time interval grows.

The criterion for selecting a model is that the model should have sufficient descriptive power to reproduce all important phenomena for the intended application, and at the same time the execution speed of a simulation should make the model tractable for use in the given application. Therefore, experiments are designed to verify whether the model has sufficient descriptive power to reproduce the necessary phenomena for traffic control, and whether the computation speed of the model is high enough. The experiment results illustrate that the higher the sampling frequency is, the more detailed the discrete model will be, but also the more computation time is needed. Hence, a trade-off can be made between the computation time and the accuracy by selecting a proper sampling time interval.

In the following chapters, model-based urban traffic controllers will be established and tested for large-scale urban traffic networks based on the models described in this chapter.



## Chapter 5

# Subnetwork MPC Controllers

Due to the high concentration of population and economic activities, a lot of traffic congestion arises in urban areas. Therefore, traffic management systems are installed to improve the performance of the existing urban transportation infrastructure, and thus to alleviate traffic congestion. Network-wide coordinated traffic management systems, which automatically and in real time determine appropriate control strategies based on the current and future traffic conditions, provide an effective control approach for improving the performance of the transportation services in cities.

A complex large-scale urban traffic network is usually divided into many small subnetworks due to the reasons mentioned in Chapter 3. These traffic subnetworks will be controlled and coordinated under a well-defined control structure. No matter which coordination scheme is used for the control structure, an advanced subnetwork controller is the foremost thing to be considered. In this chapter, a framework is given to establish MPC (Model Predictive Control) controllers for urban road subnetworks. Traffic models with different modeling details, as discussed in Chapter 4, are adopted as prediction models for the subnetwork MPC controllers. By selecting a proper prediction model, the real-time feasibility of the MPC controller can be improved a lot in practice<sup>1</sup>.

### 5.1 Introduction

Model-based control methods (including Model Predictive Control, MPC) use a prediction model and optimization in order to find the best control decisions for the network. There are already many model-based control strategies developed for urban traffic. In the 1980s and 1990s, a number of model-based optimization control strategies based on simple traffic models emerged: OPAC [49], PRODYN [44], CRONOS [17], and RHODES [115]. The model used in these control approaches are mainly simple traffic flow forecasting models based on the traffic data measured by upstream detectors. After that, model-based control strategies (including MPC) were developed based on more detailed traffic models [3, 42, 121], and they obtained good control effects. The detailed traffic models are able to describe and predict the traffic flow dynamics in the future. These model-based control

---

<sup>1</sup>The content of this chapter was published in [83].

approaches all share a similar control framework, which contains model-based prediction, on-line optimization, and rolling time horizon. The prediction model enables the controller to look ahead into the future to avoid myopic decisions. Both multiple intersections and multiple control measures can be easily coordinated through such a model-based optimization. Moreover, by using the rolling horizon procedure, feedback is introduced, which makes the controllers more robust to disturbances and model mismatch errors. All these advantages make the model-based control methods very attractive. However, the real-time feasibility<sup>2</sup> is the most common practical issue encountered when implementing MPC in practice.

When the number of controlled intersections gets larger, the optimization problems of model-based control strategies (including MPC) become too computationally complex to be solved on-line. To improve the real-time feasibility, the following methods can be considered. First, dividing the network into small subnetworks, and building distributed controllers [18, 39, 48, 70, 97]. Second, solving the optimization problem off-line, such as optimizing a feedback regulator off-line and using it with real-time measured traffic states to derive control decisions [3, 104]. Third, approximating the optimization problem by one that can be solved more efficiently. Most of the previously mentioned control strategies end up with controlling the network in a distributed structure. In this chapter, we mainly focus on the third approach. In particular, we simplify the traffic prediction model to reduce the on-line computation time. Given the initial traffic states, traffic demands, and future control decisions, any model that can predict the future traffic states of the urban traffic network, can theoretically be used as a prediction model for MPC controllers. But, different models provide different levels of modeling detail and may yield a different computational complexity. It is very important for a prediction model to offer a good trade-off between accuracy and computational complexity, so that it can be fast enough for controlling large-scale networks, while at the same time also guaranteeing effective control. Therefore, macroscopic traffic models, which do not describe the details of individual vehicles, but use more aggregated values like traffic flows and traffic densities, are suitable for real-time control purposes. Two macroscopic urban traffic network models have already been presented in Chapter 4, and we will use them as prediction model for MPC controllers. Between the two models, the BLX model is more detailed, but more complex in computation, while the S model is a simplified model that is proposed aiming at improving the real-time feasibility of the controller. This model is much faster than the previous model, while only losing a limited amount of accuracy.

## 5.2 Model Predictive Control: General framework

Model Predictive Control (MPC) [23, 92, 108] is a methodology that implements and repeatedly applies optimal control in a rolling horizon way. As Fig. 5.1 shows, in each control step, an optimal control problem is solved over a prediction horizon, but only the first control sample of the optimal control sequence is implemented. Next, the horizon is shifted one sample and the optimization is restarted again with new information of the measurements. The optimization is redone based on the prediction model of the process and an estimate of the disturbances.

<sup>2</sup>Real-time feasibility means that the on-line optimization problem can be solved fast enough so that the result is found before the time at which the controller should generate the next control signal.



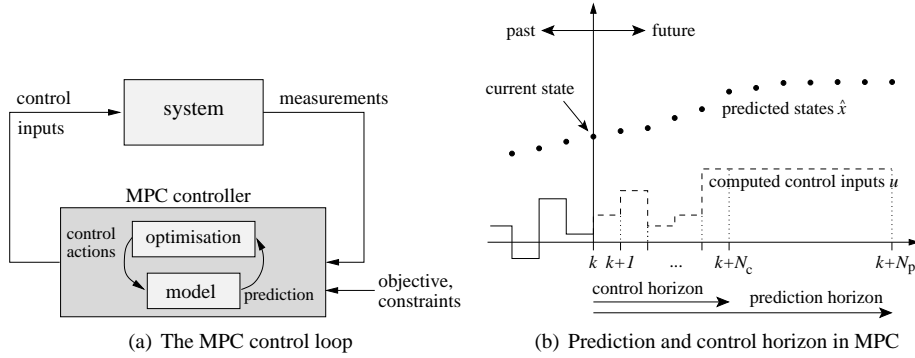


Figure 5.1: Schematic representation of MPC

In all MPC methods, three important items are recognizable in the design procedure:

- **Prediction model:** A model for the controlled system that can predict the future states. The model should be able to predict the future system states based on the information of current measured system states, the predicted future disturbances, and the future control inputs. The model can be either linear or nonlinear. On the basis of the model a prediction of the process signals over a specified horizon (i.e. prediction horizon  $N_p$ ) is made.
- **On-line optimization:** An optimization algorithm will be applied to compute a sequence of future control signals ( $\mathbf{u}(k)$ ) that minimizes the performance index subject to the given constraints, and  $u(k+j|k)$  is the control input at the  $j$ th control step in the future from the current control time step  $k$ . For linear models with linear constraints and a quadratic performance index the solution can be found using quadratic programming algorithms. For a nonlinear problem with nonlinear models or constraints or performance index, nonlinear non-convex optimization algorithms can be applied, such as multi-start sequential quadratic programming [106, Chapter 5], pattern search [7], genetic algorithm [36], etc.
- **Rolling horizon principle:** Predictive control uses the so-called rolling horizon principle. This means that after computation of the optimal control sequence, only the first control sample will be implemented, subsequently the horizon is shifted one sample and the optimization is restarted with new information of the measurements.

In order to reduce the computational complexity, two methods are usually applied to reduce the computational complexity of the on-line optimization:

1. Define control horizon  $N_c$  with  $N_c < N_p$ ;
2. Adopt aggregation techniques. Both methods decrease the computational complexity by reducing the number of control variables optimized.

In the first method, the vector of the control inputs is

$$\mathbf{u}(k) = [\mathbf{u}^T(k|k) \mathbf{u}^T(k+1|k) \cdots \mathbf{u}^T(k+N_c-1|k)]^T, \quad (5.1)$$

and we set

$$\mathbf{u}(k+i|k) = \mathbf{u}(k+N_c-1|k) \quad i = N_c, \dots, N_p - 1.$$

In the second method, the vector of the optimized control variables  $\mathbf{u}(k) \in \mathbb{R}^{N_p}$  is substituted by a vector  $\mathbf{v}(k) \in \mathbb{R}^s$  with a lower dimension by introducing an aggregation (or blocking) matrix  $\mathbf{H}$ :

$$\mathbf{u}(k) = \mathbf{H}\mathbf{v}(k), \quad (5.2)$$

where  $\mathbf{H} \in \mathbb{R}^{N_p \times s}$  with  $s < N_p$ . The aggregation matrix is the key factor of aggregation techniques. Different aggregation matrices allow different aggregation schemes. One of the most typical aggregation schemes is the blocking scheme [78, 110], which groups the optimized variables into several blocks, in each block the variables are set equal to each other. Actually, method 1 is a special case of method 2. By defining the blocking matrix, the number of the variables needed to be optimized is reduced, and the real-time computational complexity of the MPC controller also decreases. However, the control effect also deteriorates to some extent, because the aggregation constraint reduces the number of degrees of freedom of the optimization.

### 5.3 MPC for traffic subnetworks

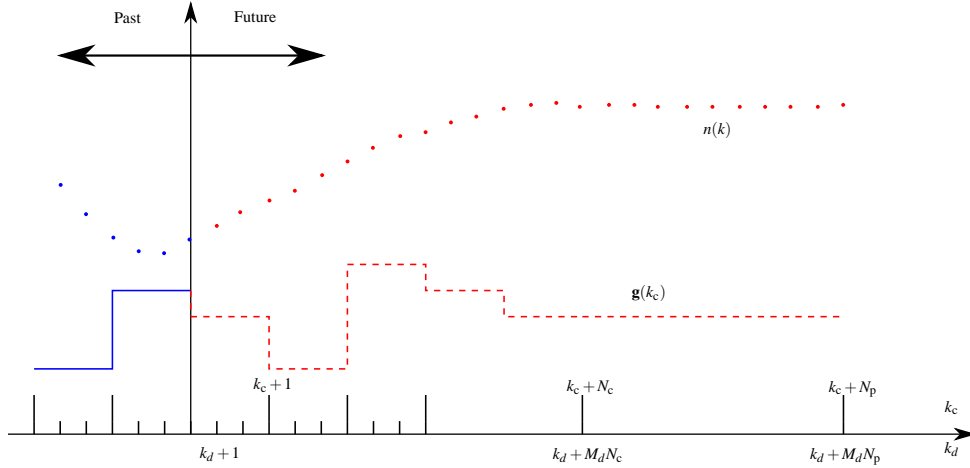


Figure 5.2: Principle of the receding horizon used in an MPC

In order to control the urban traffic subnetwork, a common control time interval needs to be defined, so that intersections within the subnetwork can communicate with each other and be synchronous. Thus, the control time interval  $T_c$  is defined as

$$T_c = N \cdot T_x, \quad (5.3)$$

where  $N$  is an integer,  $T_x$  stands for the common simulation time interval over the entire urban traffic network, which is either  $T_s$  for the BLX model (1 s, in general), or the least

common multiple time interval  $T_{\text{cm}}$  of the cycle times for all the intersections within the subnetwork for the S model. For example,

$$T_x = N_j T_j \text{ for all } j \in J, \quad (5.4)$$

as presented in Chapter 4.

For intersection  $d \in J$ , we define  $M_d = N \cdot N_d$ , then we have

$$T_c = M_d \cdot T_d, \quad (5.5)$$

where  $T_d$  is the simulation time interval for the prediction model, i.e.  $T_d = 1$  s for the BLX model, or  $T_d = c_d$  s for the S model. For any given model simulation time step counter  $k_d$  of intersection  $d \in J$ , the corresponding value of  $k_c$  can be calculated by

$$k_c(k_d) = \left\lfloor \frac{k_d}{M_d} \right\rfloor, \quad (5.6)$$

where  $\lfloor x \rfloor$  with  $x$  a real number denotes the largest integer less than or equal to  $x$ . On the other hand, a given value  $k_c$  of the control time step corresponds to the set  $\{k_c M_d, k_c M_d + 1, \dots, (k_c + 1)M_d - 1\}$  of simulation time steps for intersection  $d$ . Therefore, due to the difference between the simulation time interval and the control time interval of the urban traffic model, the traffic states  $n(k)$  are actually estimated more frequent than the variation of the control inputs (i.e. traffic signals,  $g(k_c)$ ), just as Fig. 5.2 shows.

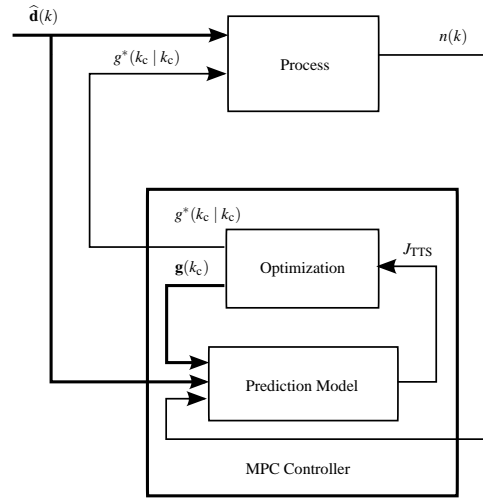


Figure 5.3: The framework of the MPC controller

The structure of the MPC controller can be illustrated by Fig. 5.3. The MPC controller obtains current traffic states  $n(k)$  from the process detectors, which can be loop detectors from real road networks. The prediction model of the MPC controller estimates the future traffic states according to the measured current traffic states  $n(k)$ , the future traffic demands  $\hat{d}(k)$ , and the given future traffic signal inputs  $g(k_c)$ . Based on this prediction model, an optimization problem can be solved, and generate a sequence of optimal traffic control

signals  $\mathbf{g}^*(k_c)$  for the future. But only the first optimal traffic control signal  $g^*(k_c|k_c)$  for the following time step is sent back to the process to be implemented in the real traffic network. Then, the time moves one step further, and all the calculations repeat in a rolling horizon way. In the following, we are going to explain the MPC control strategy in more detail.

### 5.3.1 Prediction model

The prediction models can be selected as the spatiotemporal discrete urban traffic models in Chapter 4. These models can predict the future traffic states used for evaluating the objective function based on the information of current measured traffic states, the predicted future traffic demands (i.e. future input traffic flows to the network), and the future control inputs. These traffic models, including the BLX model and the S model, can be generally described as

$$n_{u,d}(k_d + 1) = f(n_{u,d}(k_d), g_d(k_c), d_{u,d}(k_d)) \quad \text{for all } (u, d) \in L \quad (5.7)$$

where  $n_{u,d}(k_d)$  is the traffic state (e.g. the number of vehicles in a link at simulation time step  $k_d$ ), which is estimated for evaluating the objective function;  $d_{u,d}(k_d)$  is the predicted disturbance (or the traffic demand) for link  $(u, d)$  at time step  $k_d$ , which is specifically the future input traffic flow rate to the subnetwork;  $g_d(k_c)$  is the future control input of intersection  $d$ , e.g. the green times. The future traffic demands can be estimated according to the historical data, or provided by the neighbor subnetworks.

### 5.3.2 Optimization problem

Given a prediction horizon  $N_p$ , the future traffic states for link  $(u, d)$  are predicted at simulation time step  $k_d$  as

$$\hat{\mathbf{n}}_{u,d}(k_d) = [\hat{n}_{u,d}(k_d + 1|k_d) \hat{n}_{u,d}(k_d + 2|k_d) \cdots \hat{n}_{u,d}(k_d + M_d N_p|k_d)]^T, \quad (5.8)$$

based on the predicted traffic demands for link  $(u, d)$  at simulation time step  $k_d$

$$\hat{\mathbf{d}}_{u,d}(k_d) = [\hat{d}_{u,d}(k_d|k_d) \hat{d}_{u,d}(k_d + 1|k_d) \cdots \hat{d}_{u,d}(k_d + M_d N_p - 1|k_d)]^T, \quad (5.9)$$

and the future traffic control inputs for node  $d$  at control step  $k_c$

$$\mathbf{g}_d(k_c) = [g_d^T(k_c|k_c) g_d^T(k_c + 1|k_c) \cdots g_d^T(k_c + N_p - 1|k_c)]^T, \quad (5.10)$$

where  $g_d(k_c + j|k_c)$  denotes the control input at the  $j$ th control step in the future from the current control time step  $k_c$ . Assume without loss of generality that node set  $J = \{1, 2, \dots, \Phi\}$  for the traffic subnetwork, then the optimized control input for the subnetwork is a vector expressed as  $\mathbf{g}(k_c) = [\mathbf{g}_1^T(k_c) \mathbf{g}_2^T(k_c) \cdots \mathbf{g}_\Phi^T(k_c)]^T$ . Therefore, the optimization problem of MPC can be expressed as

$$\begin{aligned} \min_{\mathbf{g}(k_c)} J &= \min_{\mathbf{g}(k_c)} \sum_{(u,d) \in L} J_{u,d}(\hat{\mathbf{n}}_{u,d}(k_d), \mathbf{g}_d(k_c)) \\ \text{s.t. } \hat{n}_{u,d}(k_d) &= n_{u,d}(k_d); \\ \hat{n}_{u,d}(k_d + j + 1) &= f(\hat{n}_{u,d}(k_d + j), g_d(k_c(k_d + j)), d_{u,d}(k_d + j)), \end{aligned}$$

$$\begin{aligned}
& \text{for } j = 0, \dots, M_d N_p - 1, \text{ for all } (u, d) \in L; \\
& \Phi(\mathbf{g}(k_c)) = 0 \quad (\text{cycle time constraints}); \\
& \mathbf{g}_{\min} \leq \mathbf{g}(k_c) \leq \mathbf{g}_{\max},
\end{aligned} \tag{5.11}$$

where  $J_{u,d}$  is the objective function for link  $(u, d)$ ,  $k_c(k_d)$  is given by (5.6), and cycle time constraints guarantee that the sum of all the green time durations for an intersection equals the given cycle time. However, since the prediction model is nonlinear, and the optimization problem is computed on-line, the time taken to solve it is usually a big issue for MPC. To decrease the on-line computational complexity, a control horizon  $N_c$  ( $N_c < N_p$ ) can be defined, such that

$$g_d(k_c + i|k_c) = g_d(k_c + N_c - 1|k_c) \text{ for } i = N_c, \dots, N_p - 1. \tag{5.12}$$

### 5.3.3 Rolling horizon

Once the optimal control input  $\mathbf{g}^*(k_c)$  is derived from the optimization, then the first sample of the optimal results, i.e.

$$\mathbf{g}^*(k_c|k_c) = [g_1^{*T}(k_c|k_c) \ g_2^{*T}(k_c|k_c) \ \dots \ g_{\Phi}^{*T}(k_c|k_c)]^T, \tag{5.13}$$

is transferred to the process and implemented. When arriving to the next control step  $k_c + 1$ , the prediction model is fed with real measured traffic states, the whole prediction horizon is shifted one step forward, and the optimization starts over again. This rolling horizon scheme closes the control loop, enables the system get feedback from the real traffic network, and makes the MPC controller robust to the uncertainty and disturbances.

**Remark 5.1** The objective function in (5.11) can be selected as the Total Time Spent (TTS), Total Delay Time (TDT), Total Emission (TE), etc. (See Chapter 3) However, in the case studies of this chapter, TTS of the subnetwork is chosen as the objective function, i.e.

$$\min_{\mathbf{g}(k_c)} J = \min_{\mathbf{g}(k_c)} \sum_{(u,d) \in L} \sum_{k_d=k_c M_d+1}^{(k_c+N_p)M_d} T_d \cdot \hat{n}_{u,d}(k_d). \tag{5.14}$$

□

## 5.4 Case studies

MPC has a comparatively high requirement for on-line computational complexity. The on-line computational complexity can be decreased by increasing the efficiency, i.e. the computational speed of the prediction model. Therefore, macroscopic models are selected as the prediction models, within which the S model is proposed to further reduce the computational time. In this section, we are going to design experiments to show whether or not the selected prediction model is fast, and meanwhile also accurate enough for control purposes. In addition, the experiments will also test the control performance of the designed MPC controllers. The simulated urban road subnetwork is shown in Fig. 5.4. Nodes marked as “Sx” are the source nodes where traffic flows enter and leave the subnetwork, and also where subnetworks connect with each other.

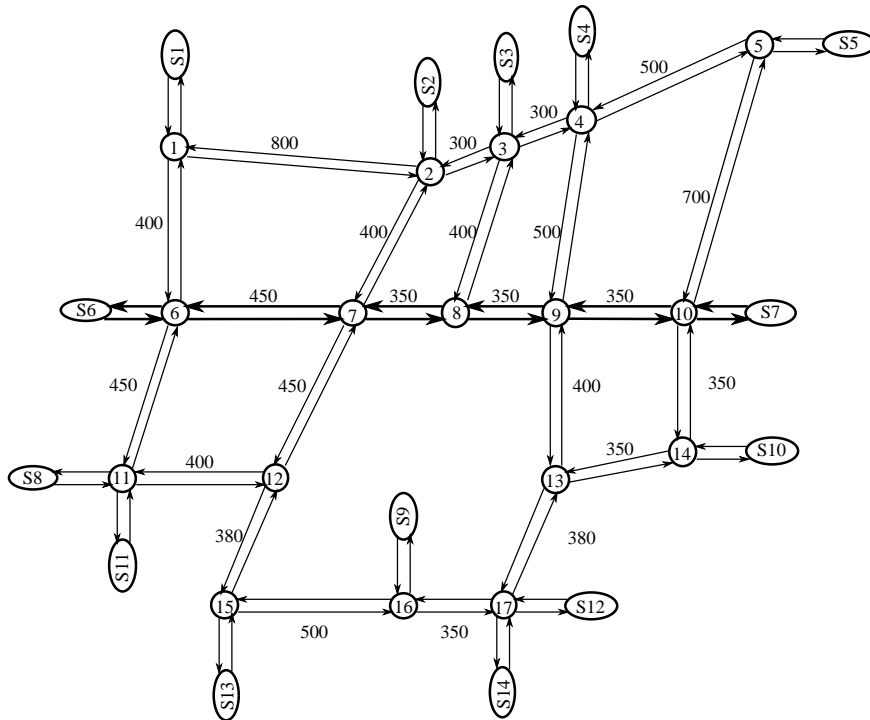


Figure 5.4: An urban road subnetwork with a main street (a string  $S6-6-7-8-9-10-S7$ )

### 5.4.1 Model test

To evaluate the effectiveness of the proposed urban traffic models, the microscopic model CORSIM developed by (FHWA, 2001) is employed to simulate the real traffic. The comparisons are performed with two measure of effectiveness defined in CORSIM, “content” and “trips”. The “content” is the cumulative count of vehicles on a link, accumulated every time step. In fact, if the simulation time step is set to 1 s in CORSIM, the “content” exactly corresponds to the TTS of a link at current simulation time instance, i.e. the performance of the controller. The “trips” is the number of vehicles that have been discharged from a link since simulation begins. Similar to the “content”, the “trips” corresponds to the total number of departure vehicles (Total Vehicle Departure) for a link, which illustrates the control effect of the control inputs (green time splits, cycle time, offset, phase) of the corresponding intersection. For control purposes, a decisive factor considered when selecting a model is whether the model can provide enough accurate relationship for the control inputs and the control outputs (performances). Thus both measure of effectiveness are chosen to evaluate the urban road subnetwork model.

The structure of a urban road subnetwork with the lengths (in meter) of the roads are shown in Fig. 5.4. In the model of this subnetwork, all the vehicle turning rates (left 33%, through 34%, right 33%), the number of lanes (3 lanes for each link) and the storage capacities of the links are considered to be fixed and known. The free-flow speed is 30 km/h. The network input flow rates of the subnetwork are all set to 2000 veh/h, which are constant in

time. A fixed-time control strategy is executed in this subnetwork, where the phases, the cycle times, and the green time splits are all constant, and the offsets are set to be zero. The cycle time is 60 s for intersection 6, 8, 9, 10, and 11, and is 40 s for the other intersections. The fixed-time signals are designed based on the data for the saturated scenario [101], i.e. the green times are proportional to the traffic demands from each direction, which depend on the saturated flow rates and the turning rates under the saturated scenario. Define  $r_p$  as the maximum saturation flow rate of phase  $p$  as

$$r_p = \max_{l \in L_p} \{\mu_l\}, \quad (5.15)$$

where  $L_p$  is the set of lanes of phase  $p$ , and  $\mu_l$  is saturation flow rate for lane  $l$ . Then, define  $R$  as the summation of all maximum saturation flow rates corresponding to each of the  $n$  phases in one cycle:

$$R = \sum_{p=1}^n r_p. \quad (5.16)$$

For each phase  $p$ , its optimum green time,  $t_p^{\text{green}}$ , is calculated by distributing the total available green time, i.e.  $C - Y$  ( $C$  is the cycle time length,  $Y$  is the total yellow time length), in proportion to its saturation flow rates of the corresponding directions:

$$t_p^{\text{green}} = \frac{r_p}{R} (C - Y). \quad (5.17)$$

Fig. 5.5 shows the comparisons of the S model, BLX model, and CORSIM on the indices of TTS and total vehicle departure for two links in the network of Fig. 5.4. As Fig. 5.5 shows, even though they are macroscopic models, the BLX model and S model are able to provide curves of the two measure of effectiveness that are consistent with that of microscopic traffic simulator, CORSIM, for both links. But the curves of the BLX model and S model will drift away from the curves of CORSIM when the time grows. The reason is that the longer time the models run, the more errors will be accumulated for the macroscopic models because of neglecting the detailed driving behavior of individual vehicles compared with a microscopic model. The figures also show that the modeling accuracy is higher for the link near the source nodes of the network (i.e. link (10,9)) than for the link inside the network (i.e. link (9,8)). This is because that the farther away a link is located from the network boundary, the more errors will be accumulated, as they are passed down from the upstream links.

The evolution of the average number of departure vehicles leaving from link (9,8) and link (10,9) are shown in Fig. 5.6. The S model is obtained through model reduction from the more detailed BLX model, and thus it sacrifices some modeling accuracy to gain more computation efficiency. But, Fig. 5.5 illustrates that the S model has similar curve trends as the BLX model in both the CORSIM measure of effectiveness at the two links. Similar experiment results can be derived for the other links within the network. Therefore, this experiment illustrates that both the macroscopic models, the S model and the BLX model, are accurate enough to be selected as the prediction model of MPC controllers in this case.

When used as prediction models for MPC, the computational speed is also very important. The experiments above were also used to compare the speed for running a simulation by using CORSIM, the BLX model, and the S model. The three models are provided with the same traffic network, the same network parameters and traffic demands as the previous

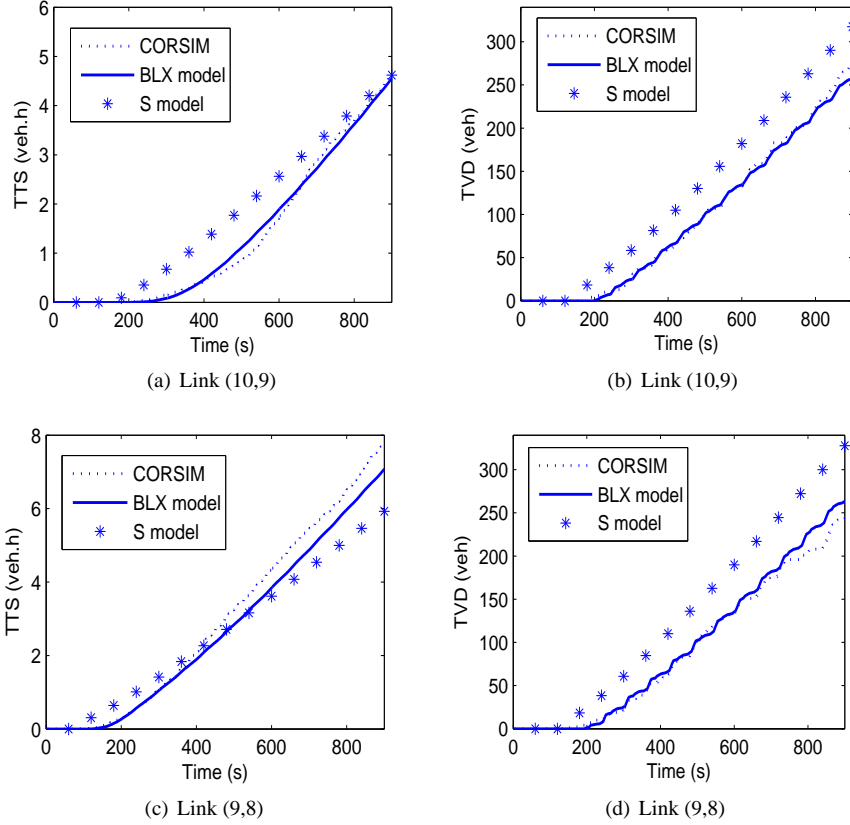


Figure 5.5: TTS and Total Vehicle Departure (TVD) comparison of the S model, BLX model, and CORSIM for different links

experiment set-up, and simulate for a same period of time, 30 min. The results turn out to be that the BLX model and the S model reduce the computational time by 70.8 % and 99.2 % respectively compared to CORSIM. The S model reduces the computational time by 97.4 % compared to the BLX model. Therefore, as macroscopic models, both the BLX model and the S model are faster and in this context more suitable as prediction model for the MPC controller than the CORSIM simulator. Moreover, the S model is much faster than the BLX model, which can further increase the on-line feasibility of MPC controller, but with a limited loss of the control performance.

#### 5.4.2 Urban subnetwork control using MPC

Two MPC controllers are designed for the urban road subnetwork shown in Fig. 5.4, taking the BLX model and the S model as prediction models respectively. The structure of the BLX-based MPC controller and the S-based MPC controller is shown in Fig. 5.7. The fixed-time (FT) controller is simulated as a benchmark to evaluate and compare these two



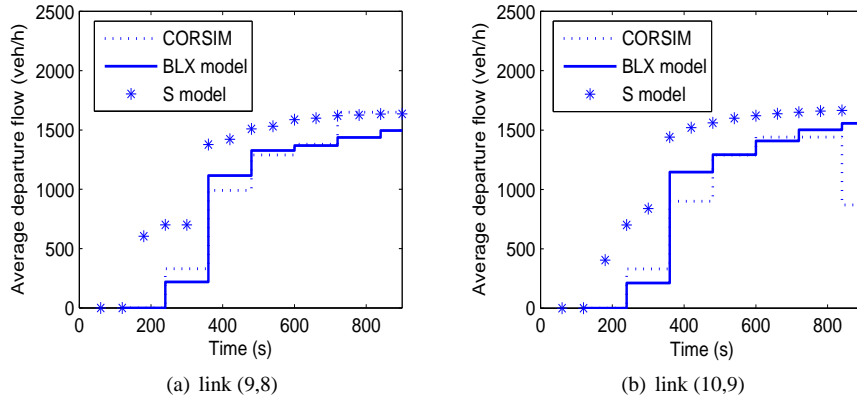


Figure 5.6: The evolution of the average number of departure vehicles from CORSIM, BLX, and S model

control strategies. In order to evaluate these control strategies, CORSIM is used to simulate the real traffic environment.

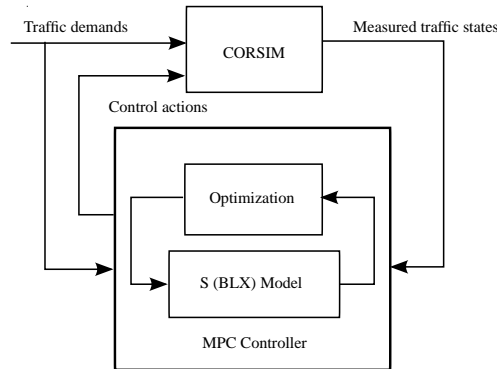


Figure 5.7: Illustration of MPC controllers

The same experiment set-up of the previous subsection is applied again for this subsection. During the experiments, the simulation time interval of the BLX model is set to 1 s, while in the S model, the simulation time intervals are 60 s or 40 s. For both the MPC controllers, the control time interval  $T_c$  is 120 s, the prediction horizon  $N_p$  is 5, and  $N_u = N_p$ . All the simulations implemented with different control strategies run for the same time period, 1 h. Total Time Spent (TTS) is the control objective of the MPC controllers, but the performance indicators used for the evaluation are selected as the TTS and the Total Delay Time (TDT). TTS is the accumulated amount of time spent by all the vehicles inside the road network since the beginning of the simulation, including both the vehicles freely running on a link and the vehicles slowing down or waiting in queues. TDT is the total time spent by all the vehicles traveling with speed lower than the free-flow speed inside the road

network since the beginning of the simulation, i.e. the total amount of time that the vehicles are delayed. Using extra performance indicator TDT, traffic delays can be further evaluated and compared for the urban traffic network. Two traffic scenarios are considered:

1. *Balanced scenario*: The traffic demands (traffic flows) from all the source nodes into the subnetwork are the same, and they all increase with time as Fig. 5.8 shows.
2. *Imbalanced scenario*: The traffic demands of all the source nodes are very low (500 veh/h), but the traffic demands for source nodes S6 and S7 are very high (3000 veh/h). Therefore, the road between S6 and S7 becomes a busy and main street of the subnetwork.

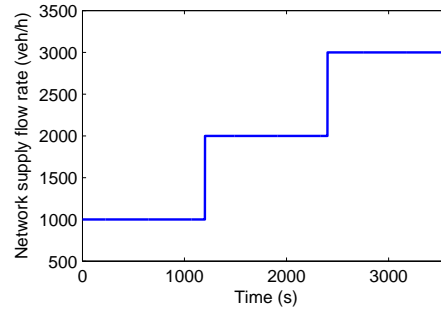


Figure 5.8: The variation of the supply flow rates for the network

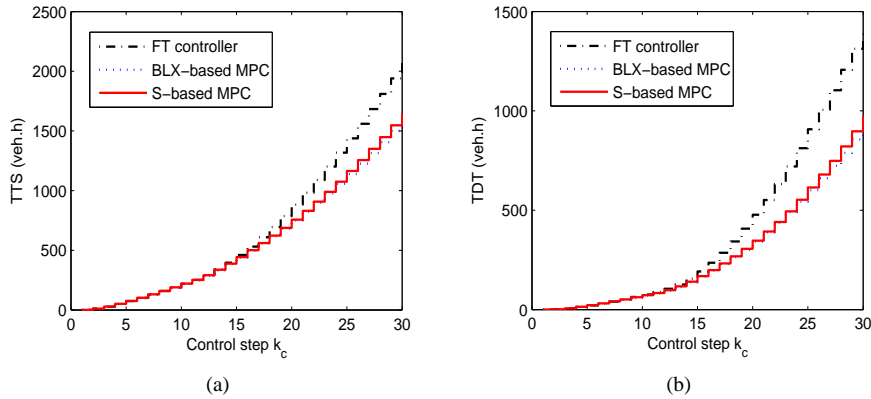


Figure 5.9: TTS and TDT comparisons for the subnetwork in Fig. 5.4 of the S-based MPC, the BLX-based MPC, and the fixed-time controller at every control time step in the balanced scenario

For the balanced scenario, Fig. 5.9 illustrates the comparison of the two control performance indicators for all the control strategies (S-based MPC, BLX-based MPC, and FT

controller). As Fig. 5.9(a) shows, the three controllers can control the subnetwork to almost the same value of the TTS at the beginning of the simulation. It means that all the three controllers are almost equally effective, when the traffic demands are low, and the subnetwork has a comparatively low traffic density. This is because, if the cycle times of intersections and off-sets between intersections are fixed, the control performance mainly depends on the traffic demands rather than on the traffic signal splits when the subnetwork is far from saturated. However, as the traffic demands increase, and the traffic density of the subnetwork grows, the MPC controllers become more and more superior to the fixed-time controller. Even though the TTS is selected as the control objective for the MPC controllers, the TDT is reduced correspondingly, as Fig. 5.9(b) shows. In order to further improve the control performance in unsaturated scenario, the cycle time lengths and the traffic signal off-sets can be further optimized. However, the computational complexity will also increase correspondingly, as the number of the optimized variables grows. To avoid increasing the computational complexity, we can leave the responsibility of optimizing and deciding the cycle time lengths and off-sets to the higher level controller (i.e. the supervisor or the coordinator).

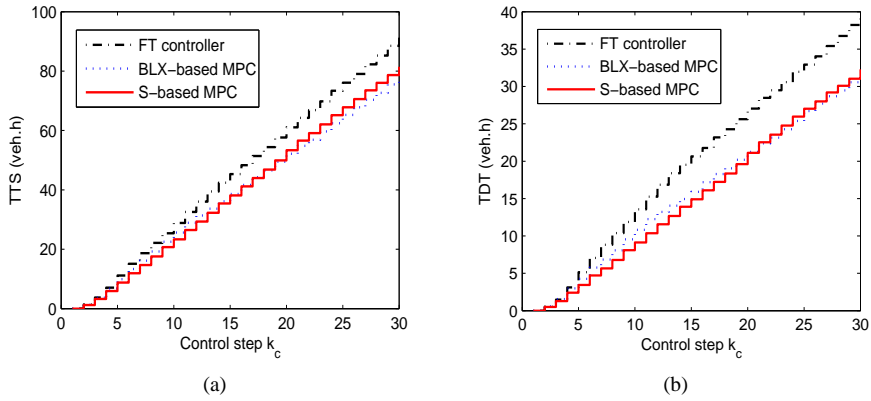


Figure 5.10: TTS and TDT comparisons for the string in Fig. 5.4 among the S-based MPC, the BLX-based MPC, and the fixed-time controller at every control time step in the unbalanced scenario

For the unbalanced scenario, both the TTS and TDT are reduced obviously for the string S6-6-7-8-9-10-S7 in Fig. 5.4 by the MPC controllers compared with the fixed-time controller, as Fig. 5.10 shows. In average, the TTS is reduced 6.3%, and the TDT is reduced 14.1%. However, TTS and TDT for the other links of the subnetwork except the string keep the same as that of the fixed-time controller, and sometimes are even worse, as Fig. 5.11 shows. This means that, in order to reduce the delays on the string, the MPC controllers hold the traffic flows back on the other links in the subnetwork. In such unbalanced situation, the MPC controllers coordinate the traffic signals within the subnetwork by sacrificing certain performances of the links that is less crowded, so as to achieve a better network overall performance. Comparing Fig. 5.9 and Fig. 5.10, the MPC controllers are able to balance the traffic flow distribution and coordinate the control measures within the subnet-

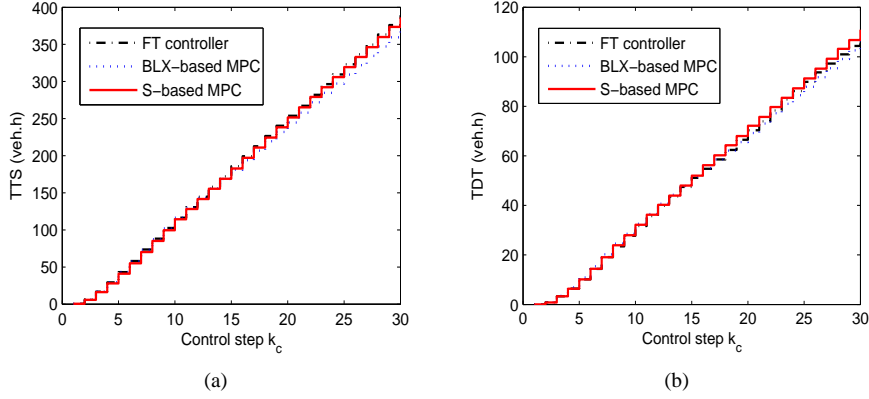


Figure 5.11: TTS and TDT comparisons for other links except the string among the S-based MPC, the BLX-based MPC, and the fixed-time controller at every control time step in the unbalanced scenario

work. Therefore, the coordination function of the MPC controllers is more obvious in the unbalanced scenario than in the balanced scenario.

Fig. 5.12 shows the evolution of the average number of vehicles in link (6,7) and (7,8) for S-based MPC, BLX-based MPC, and the fixed-time controller in both the balanced and the imbalanced scenario. As the figure shows, there are no spillbacks in the two links in either the balanced scenario or the imbalanced scenario. But, the vehicles controlled by MPC controllers spread more evenly than the vehicles controlled by the FT controller, which make better use of the capacity of the traffic network, and result in better overall control performance.

In the previous simulation, we can derive that the CFL condition in Chapter 4 is not always kept. Therefore, a new simulation was run for the case study, in which the CFL condition is not violated. All the setups are the same as the previous simulation, except the traffic network is as Fig. 5.14 shows, new supply flow rates for the network are changed as Fig. 5.13 shows to create a peak for the traffic supply, and the cycle time becomes 120 s for Intersection 6, and 60 s for the other intersections.

The TTS curves are very similar for the fixed-time controller, BLX-based MPC, and S-based MPC in the balanced scenario, as Fig. 5.15 shows. In the imbalanced scenario, the string allows more traffic flow under the MPC controllers compared with the fixed-time controller, and alleviates the traffic burden in the rest of the network (see Fig. 5.16 and Fig. 5.17).

To on-line solve the nonlinear optimization problem in (5.11) for the MPC controller, the prediction model has to be simulated thousands of times. The speed for solving the optimization problem can be significantly improved by reducing the computational speed of the prediction model. The on-line optimization problem of the S-based controller can be solved within 1 min for the subnetwork of Fig. 5.4, and the cpu time can be reduced around 96% compared to the BLX-based controller. Thus, the S-based controller requires much less on-line computational time than the BLX-based controller because of the effi-

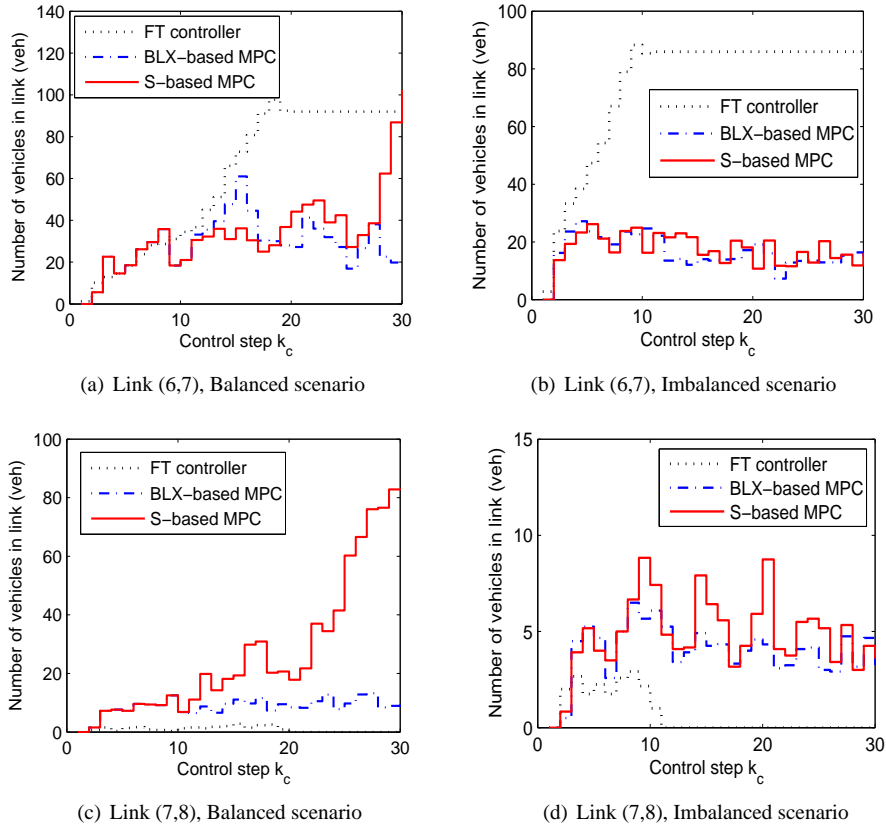


Figure 5.12: The evolution of the average number of vehicles in links (6,7) and (7,8) for S-based MPC, BLX-based MPC, and the fixed-time controller

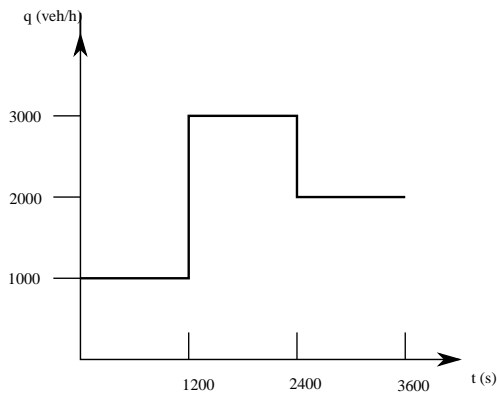


Figure 5.13: The variation of the supply flow rates for the network

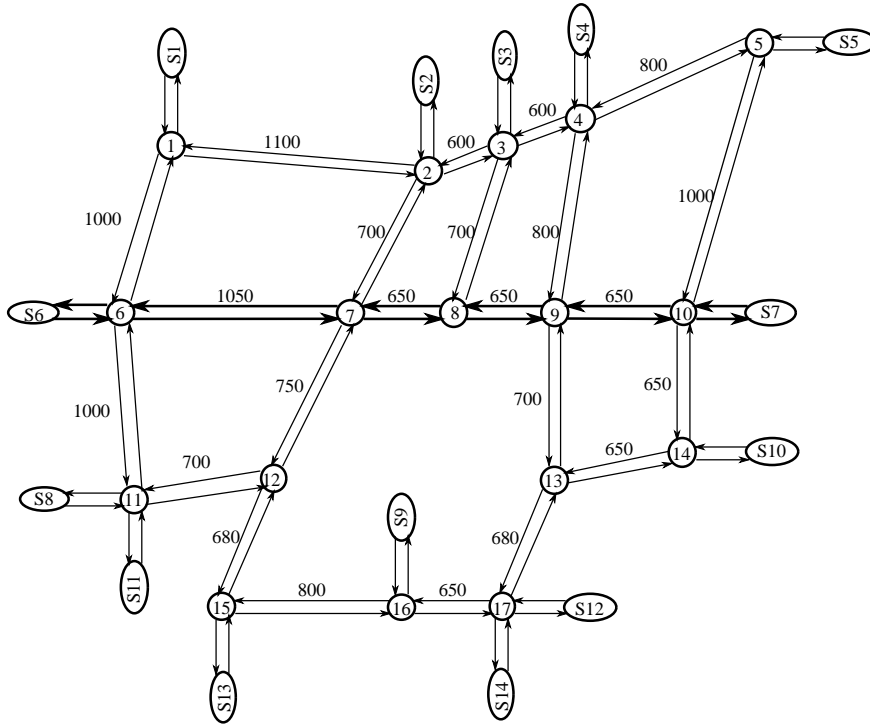


Figure 5.14: An urban road subnetwork with a main street (a string  $S_6-6-7-8-9-10-S_7$ )

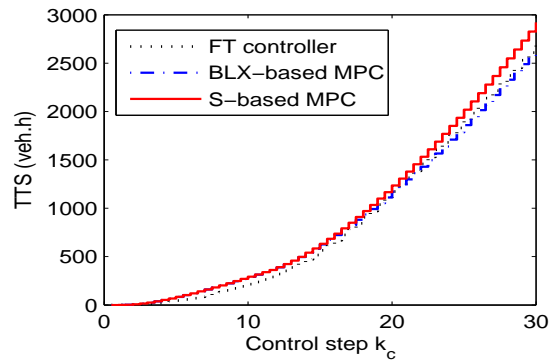


Figure 5.15: TTS comparisons for the subnetwork in Fig. 5.14 for S-based MPC, BLX-based MPC, and the fixed-time controller at every control time step in the balanced scenario

ciency of the S model. Moreover, it is able to keep a similar performance as the BLX-based controller. Therefore, the real-time computational feasibility of the MPC controller can be greatly improved by using the S model as the prediction model, but without losing much

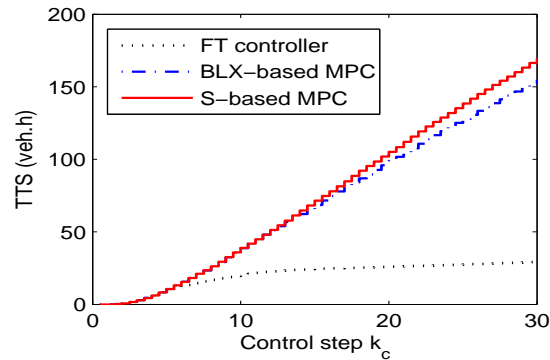


Figure 5.16: TTS comparisons for the string in Fig. 5.14 for S-based MPC, BLX-based MPC, and the fixed-time controller at every control time step in the unbalanced scenario

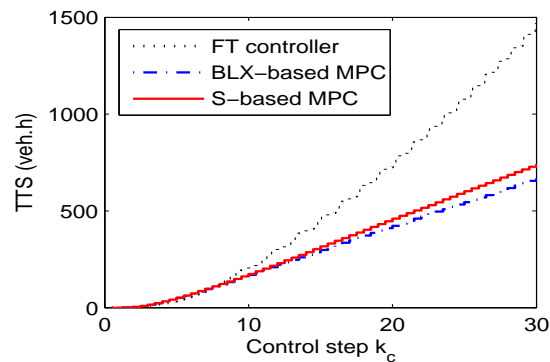


Figure 5.17: TTS comparisons for the other links except the string in Fig. 5.14 for S-based MPC, BLX-based MPC, and the fixed-time controller at every control time step in the unbalanced scenario

performance.

In theory, it is possible to solve an MPC problem with 100 intersections. However, the on-line computational complexity will almost increase exponentially with the expansion of the network scale. Therefore, it may not be practical to control 100 intersections with the current computing capability. But, we could divide the large network into subnetworks, and apply hierarchical or distributed control structures as suggested in Chapter 3.

## 5.5 Summary

As an advanced control methodology, MPC has many advantages, like robustness to disturbances, long-term sight, easy dealing with constraints, and so on. However, despite of all

these advantages, it also inevitably gives rise to the problem of high on-line computational complexity. In this chapter, an efficient but also effective MPC controller for urban road subnetworks is presented. The efficiency of the subnetwork controllers is also the basis of an efficient coordinating algorithm for the subnetworks.

To improve the applicability of the MPC controller in practice, the characteristics of the prediction models are considered. Two macroscopic models, the S model and the BLX model, described in Chapter 4, are considered as the prediction models of MPC controllers. To further increase the computational speed, the S model, which is a reduced version of the BLX model, is selected. Simulation results show that both the S model and the BLX model are suitable as prediction model of MPC, and that the S model is much faster while still offering acceptable accuracy in the predictions. MPC controllers taking the two models as prediction model respectively are constructed and investigated. The MPC controllers show great capability for coordinating the traffic measurements and intersections within the subnetworks and achieve a good overall performance, especially in the situation that the traffic flows are not uniformly scattered within the roads of subnetwork. From a computational point of view, the S model-based MPC controller is much more efficient than the BLX model-based MPC controller, while only incurring a limited reduction of the control performance.



## Chapter 6

# Fast MPC for Urban Traffic Subnetworks via MILP

In this chapter, we are going to investigate another method to increase the real-time feasibility of the on-line optimization problems, when we apply MPC to control large-scale urban traffic networks. The presented method is to reformulate the on-line optimization problem into another format, so that it can be solved more efficiently than before<sup>1</sup>.

### 6.1 Introduction

Due to the nonlinear nature of the prediction model (e.g. the models discussed in Chapter 4), the optimization problem of MPC for urban traffic networks is a nonlinear non-convex optimization problem. As a consequence, the on-line computational complexity becomes a big challenge for the MPC controller, if it is implemented in real-life traffic network. One can reduce the on-line computational complexity through model reduction (as the S model in Chapter 4), but the degree that the computation speed can be improved is still limited due to the nonlinear non-convex nature of the optimization problem. Therefore, in this chapter, we mainly focus on improving the real-time feasibility of MPC controllers through improving the efficiency of solving optimization problems.

In general, a nonlinear non-convex optimization problem needs to be solved by nonlinear optimization algorithms, e.g. multi-start Sequential Quadratic Programming (SQP) [106, Chapter 5], Pattern Search [7], Genetic Algorithms [36], and Simulated Annealing [43]. Among them, SQP is local optimization method, while others are global optimization methods. But, through selecting multiple initial starting points, and choosing the best solution, multi-start SQP is able to refine the solution and approach the global optimum. However, all the optimization methods mentioned above require a huge number of evaluations of the objective function, which results in running the prediction model a huge number of times. Although the S model is fast already, it still takes quite some time to simulate the model repeatedly. Therefore, the optimization problem inevitably suffers from an exponentially growing computational complexity when the scale of the controlled traffic

---

<sup>1</sup>The content of this chapter was published in [84, 85, 88].

network grows. Consequently, the MPC controller will become real-time infeasible when the scale of the controlled traffic network grows. Therefore, we will reformulate this nonlinear non-convex optimization problem into an optimization problem that can be solved more efficiently.

To this aim, the nonlinear macroscopic urban traffic network model, the S model of Chapter 4, i.e. the prediction model of the MPC controller, is reformulated and linearized into linear equalities and inequalities by introducing in auxiliary integer variables. Based on this reformulated model, the original nonlinear non-convex on-line optimization problem of the MPC controller is rewritten into a Mixed-Integer Linear Programming (MILP) problem, which can be solved efficiently by an existing MILP solver. Thereafter, the real-time feasibility of the MPC control strategy can be further increased. Moreover, we propose an approach to reduce the complexity of the MILP optimization problem even further. The simulation results show that the MILP-based MPC controllers can reach the same performance as the original MPC controller, but the time taken to solve the optimization becomes only a few seconds, which is a significant reduction compared with the time required by the original MPC controller.

## 6.2 Rules for equivalent transformation into MLD model

### 6.2.1 Preliminaries

First, some basic tools are introduced for transforming logical statements involving continuous variables into mixed-integer linear inequalities<sup>2</sup>.

Capital letters  $X_i$  are used to represent statements, e.g. “ $x \leq 0$ ” or “Color is black”.  $X_i$  is commonly referred to as a literal, and has a truth value of either “T” (true) or “F” (false). Boolean algebra enables statements to be combined in compound statements by means of connectives:

- “ $\wedge$ ” — and;
- “ $\vee$ ” — or;
- “ $\sim$ ” — not;
- “ $\Rightarrow$ ” — implies;
- “ $\Leftrightarrow$ ” — if and only if;
- “ $\oplus$ ” — exclusive or.

These connectives are defined by means of the truth table given in Table 6.1. The following properties will be used later on:

$$X_1 \Rightarrow X_2 \text{ is the same as } \sim X_1 \vee X_2 \quad (6.1)$$

$$X_1 \Rightarrow X_2 \text{ is the same as } \sim X_2 \Rightarrow \sim X_1 \quad (6.2)$$

$$X_1 \Leftrightarrow X_2 \text{ is the same as } (X_1 \Rightarrow X_2) \wedge (X_2 \Rightarrow X_1). \quad (6.3)$$

<sup>2</sup>This subsection is based on [9, 37].

Table 6.1: Truth table

$X_1$	$X_2$	$X_1 \wedge X_2$	$X_1 \vee X_2$	$\sim X_1$	$X_1 \Rightarrow X_2$	$X_1 \Leftrightarrow X_2$	$X_1 \oplus X_2$
T	T	T	T	F	T	T	F
T	F	F	T	F	F	F	T
F	T	F	T	T	T	F	T
F	F	F	F	T	T	T	F

One can associate with a literal  $X_i$  a logical variable  $\delta_i \in \{0, 1\}$ , which has a value of either 1 if  $X_i = T$ , or 0 if  $X_i = F$ . The following propositions and linear constraints are then equivalent:

$$X_1 \wedge X_2 \quad \text{is equivalent to} \quad \delta_1 = \delta_2 = 1 \quad (6.4)$$

$$X_1 \vee X_2 \quad \text{is equivalent to} \quad \delta_1 + \delta_2 \leq 1 \quad (6.5)$$

$$\sim X_1 \quad \text{is equivalent to} \quad \delta_1 = 0 \quad (6.6)$$

$$X_1 \Rightarrow X_2 \quad \text{is equivalent to} \quad \delta_1 - \delta_2 \leq 0 \quad (6.7)$$

$$X_1 \Leftrightarrow X_2 \quad \text{is equivalent to} \quad \delta_1 - \delta_2 = 0 \quad (6.8)$$

$$X_1 \oplus X_2 \quad \text{is equivalent to} \quad \delta_1 + \delta_2 = 1. \quad (6.9)$$

We can use this computational inference technique to model logical parts of processes (on/off switches, discrete mechanisms, combinational and sequential networks) and heuristics knowledge about plant operation as integer linear inequalities. In this way, we can construct models of hybrid systems.

## 6.2.2 Equivalent transformation into MLD model

A Mixed Logical Dynamical (MLD) model [9, 37] allows specifying the evolution of continuous variables through linear dynamic equations, of discrete variables through propositional logic statements and automata, and the mutual interaction between the two. The key idea of transforming a model into an MLD model consists of embedding the logic part in the state equations by transforming boolean variables into 0/1 integers and by expressing the relations as mixed-integer linear inequalities.

According to [27], consider the statement  $f(x) \leq 0$ , where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ . Assume that  $x \in \mathcal{X}$ , where  $\mathcal{X} \subset \mathbb{R}^n$  is a given bounded set, and define

$$M = \max_{x \in \mathcal{X}} f(x), \quad m = \min_{x \in \mathcal{X}} f(x). \quad (6.10)$$

Theoretically, an over-estimate (or under-estimate) of  $M$  or  $m$  suffices for our purpose. However, more realistic estimates provide computational benefits. Now, by introducing in  $\delta \in \{0, 1\}$ , it is easy to verify that

$$[f(x) \leq 0] \wedge [\delta = 1] \quad \text{is true if and only if} \quad f(x) - \delta \leq -1 + m(1 - \delta) \quad (6.11)$$

$$[f(x) \leq 0] \vee [\delta = 1] \quad \text{is true if and only if} \quad f(x) \leq M\delta \quad (6.12)$$

$$\sim [f(x) \leq 0] \quad \text{is true if and only if} \quad f(x) \geq \varepsilon, \quad (6.13)$$

where  $\varepsilon$  is a small tolerance (typically the machine precision), beyond which the constraint is regarded as violated.

**Remark 6.1** The reason for introducing  $\varepsilon$  is that an equation like  $f(x) > 0$  does not fit the mixed integer linear programming framework, in which only nonstrict inequalities are allowed. Therefore, the equation  $f(x) > 0$  is replaced by the equation  $f(x) \geq \varepsilon$  with  $\varepsilon$  a small tolerance, typically the machine precision, where we assume that in practice the case  $0 < f(x) < \varepsilon$  cannot occur due to the finite number of bits used for representing real numbers on a computer.  $\square$

Then, the following equivalence holds

$$[f(x) \leq 0] \Rightarrow [\delta = 1] \quad \text{is true if and only if} \quad f(x) \geq \varepsilon + (m - \varepsilon)\delta \quad (6.14)$$

$$[f(x) \leq 0] \Leftrightarrow [\delta = 1] \quad \text{is true if and only if} \quad \begin{cases} f(x) \leq M(1 - \delta) \\ f(x) \geq \varepsilon + (m - \varepsilon)\delta \end{cases} \quad (6.15)$$

Moreover, the term  $\delta f(x)$ , where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $\delta \in \{0, 1\}$ , can be replaced by the auxiliary real variable  $z = \delta f(x)$  which satisfies  $[\delta = 0] \Rightarrow [z = 0]$ ,  $[\delta = 1] \Rightarrow [z = f(x)]$ . Therefore, by defining  $M$  and  $m$  as in (6.10),  $z = \delta f(x)$  is equivalent to

$$\begin{cases} z \leq M\delta \\ z \geq m\delta \\ z \leq f(x) - m(1 - \delta) \\ z \geq f(x) - M(1 - \delta) \end{cases} \quad (6.16)$$

### 6.3 Reformulation of the urban traffic model

Now we will show that the nonlinear non-convex optimization problem (5.10) can be reformulated into a mixed-integer linear optimization problem [84], which can be solved efficiently by existing MILP (Mixed-Integer Linear Programming) solvers [6, 46, 89]. The MILP solver is more efficient than the SQP solver for this particular optimization problem, and can find the global optimum rather than a local optimum.

#### 6.3.1 Model reformulation into mixed-integer linear model

We now show how the model (4.53) can be reformulated as mixed-integer linear equations and inequalities using the equivalent reformulation rules above. Let

$$\begin{aligned} a &= \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_d) / c_d \\ b &= (q_{u,d,o}(k_d) / c_d) + \alpha_{u,d,o}^{\text{arriv}}(k_d) \\ c &= \beta_{u,d,o}(k_d)(C_{d,o} - n_{d,o}(k_d)) / c_d \\ d &= \min(a, b), \end{aligned} \quad (6.17)$$

then (4.53) becomes

$$\alpha_{u,d,o}^{\text{leave}}(k_d) = \min(a, b, c) = \min(d, c) . \quad (6.18)$$

Let

$$f_1 = b - a, \quad (6.19)$$

and define

$$\delta_1 = \begin{cases} 1 & \text{if } f_1 \leq 0 \\ 0 & \text{if } f_1 > 0 \end{cases}, \quad (6.20)$$

where  $\delta_1 = 1$  means free flow demand, and  $\delta_1 = 0$  means queue discharge demand. Then we have

$$d = a + (b - a) \cdot \delta_1 = a + f_1 \cdot \delta_1. \quad (6.21)$$

Similarly, let

$$f_2 = c - d, \quad (6.22)$$

and define

$$\delta_2 = \begin{cases} 1 & \text{if } f_2 \leq 0 \\ 0 & \text{if } f_2 > 0 \end{cases}, \quad (6.23)$$

where  $\delta_2 = 1$  means spillback regime, and  $\delta_2 = 0$  means free entry flow regime. Then we have

$$\min(d, c) = d + (c - d) \cdot \delta_2 = d + f_2 \cdot \delta_2. \quad (6.24)$$

Let

$$z_1 = f_1 \cdot \delta_1 \quad (6.25)$$

$$z_2 = f_2 \cdot \delta_2 \quad (6.26)$$

and substitute (6.21) into (6.24), then (6.18) becomes linear, as

$$\alpha_{u,d,o}^{\text{leave}}(k_d) = a + z_1 + z_2. \quad (6.27)$$

According to the equivalent transformation rules, (6.20) and (6.25) are equivalent to the inequality constraints

$$\begin{aligned} f_1 &\leq M_1(1 - \delta_1) \\ f_1 &\geq \varepsilon + (m_1 - \varepsilon)\delta_1 \\ z_1 &\leq M_1\delta_1 \\ z_1 &\geq m_1\delta_1 \\ z_1 &\leq f_1 - m_1(1 - \delta_1) \\ z_1 &\geq f_1 - M_1(1 - \delta_1). \end{aligned} \quad (6.28)$$

Similarly, (6.23) and (6.26) are equivalent to the inequality constraints

$$\begin{aligned} f_2 &\leq M_2(1 - \delta_2) \\ f_2 &\geq \varepsilon + (m_2 - \varepsilon)\delta_2 \\ z_2 &\leq M_2\delta_2 \\ z_2 &\geq m_2\delta_2 \\ z_2 &\leq f_2 - m_2(1 - \delta_2) \\ z_2 &\geq f_2 - M_2(1 - \delta_2). \end{aligned} \quad (6.29)$$

Here,  $M_1$  and  $m_1$  are the maximum value and the minimum value of  $f_1$ , and  $M_2$  and  $m_2$  are the maximum value and the minimum value of  $f_2$ . These upper and lower bounds depend on the capacity or the saturated flow rate of link  $(u, d)$ , or the capacity of its downstream link. According to (6.17), the upper bounds and lower bounds of  $a$ ,  $b$ ,  $c$ , and  $d$  can be deduced as

$$\begin{aligned} a_{\min} &= 0 \leq a \leq \mu_{u,d} = a_{\max} \\ b_{\min} &= 0 \leq b \leq C_{u,d}/c_d = b_{\max} \\ c_{\min} &= 0 \leq c \leq C_{d,o}/c_d = c_{\max} \\ d_{\min} &= 0 \leq d \leq \min(\mu_{u,d}, C_{u,d}/c_d) = d_{\max}, \end{aligned} \quad (6.30)$$

where all the lower bounds are 0, which represents the average traffic flow rate is non-negative in reality; the upper bounds depend on the capacity of link  $(u, d)$  or its downstream link. With the upper bounds and lower bounds of  $a$ ,  $b$ ,  $c$ , and  $d$ , we can derive

$$\begin{aligned} M_1 &= b_{\max} - a_{\min} = C_{u,d}/c_d \\ m_1 &= b_{\min} - a_{\max} = -\mu_{u,d} \\ M_2 &= c_{\max} - d_{\min} = C_{d,o}/c_d \\ m_2 &= c_{\min} - d_{\max} = -\min(\mu_{u,d}, C_{u,d}/c_d). \end{aligned} \quad (6.31)$$

Therefore, by introducing the additional auxiliary binary variables  $\delta_1$  and  $\delta_2$ , and the auxiliary real variables  $f_1$ ,  $f_2$ ,  $z_1$ , and  $z_2$ , the original formula (4.53) in the urban traffic model is equivalently reformulated as linear equations (6.19), (6.22), and (6.27), and mixed-integer linear inequalities (6.28)-(6.29).

### 6.3.2 Reformulation of the model synchronization

Consider (4.63) for fixed  $i, u, d$ , and  $k_d$ . We will now show that this results in

$$\alpha_{i,u,d}^{\text{enter}}(k_d) = \mathcal{F}_{\text{in}}(\alpha_{i,u,d}^{\text{leave}}(k_u), \dots, \alpha_{i,u,d}^{\text{leave}}(k_u + \ell)), \quad (6.32)$$

with  $\ell$  an integer and  $\mathcal{F}_{\text{in}}$  a linear function.

In (4.63),  $\alpha_{i,u,d}^{\text{leave,cont}}(t)$  is a piecewise constant function with intervals  $\xi(k_u), \dots, \xi(k_u + \ell)$  and function values  $\alpha_{i,u,d}^{\text{leave}}(k_u), \dots, \alpha_{i,u,d}^{\text{leave}}(k_u + \ell)$ , where  $\xi(x)$  depends on  $c_d$ ,  $c_u$ , and  $\Delta c_{u,d}$ . Once these variables are fixed,  $\xi(x)$  is fixed. Hence, we have linear expression

$$\alpha_{i,u,d}^{\text{enter}}(k_d) = \frac{1}{c_d} \sum_{j=0}^{\ell} \xi_{k_d}(k_u + j) \alpha_{i,u,d}^{\text{leave}}(k_u + j), \quad (6.33)$$

as Fig. 6.1 shows.

The linear function  $\mathcal{F}_{\text{in}}$  can be derived by the following approach. Given (4.63), we define

$$\begin{aligned} k_u^+ &= \text{floor} \left\{ \frac{(k_d + 1) \cdot c_d + \Delta c_{u,d}}{c_u} \right\}, \\ \theta_u^+ &= \text{rem} \left\{ (k_d + 1) \cdot c_d + \Delta c_{u,d}, c_u \right\}, \\ k_u^- &= \text{floor} \left\{ \frac{k_d \cdot c_d + \Delta c_{u,d}}{c_u} \right\}, \\ \theta_u^- &= \text{rem} \left\{ k_d \cdot c_d + \Delta c_{u,d}, c_u \right\}. \end{aligned} \quad (6.34)$$

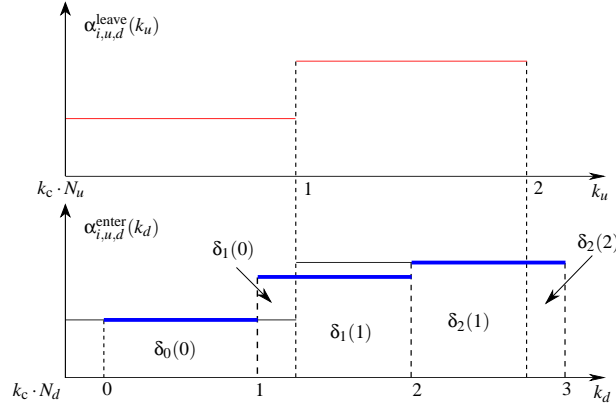


Figure 6.1: Illustration for linear intersection synchronization

where  $k_u^+ \geq k_u^-$  and  $0 \leq \theta^+ < c_u$ ,  $0 \leq \theta^- < c_u$ . Then, we obtain

$$\begin{aligned}
 \alpha_{i,u,d}^{\text{enter}}(k_d) &= \frac{1}{c_d} \int_{k_u^- c_u + \theta_u^-}^{k_u^+ c_u + \theta_u^+} \alpha_{i,u,d}^{\text{leave,cont}}(t) dt \\
 &= \frac{1}{c_d} \left[ \int_{k_u^- c_u + \theta_u^-}^{(k_u^- + 1)c_u} \alpha_{i,u,d}^{\text{leave}}(k_u^-) dt \right. \\
 &\quad + \sum_{i=1}^{k_u^+ - k_u^- - 1} \int_{(k_u^- + i)c_u}^{((k_u^- + i + 1)c_u} \alpha_{i,u,d}^{\text{leave}}(k_u^- + i) dt \\
 &\quad \left. + \int_{k_u^+ c_u}^{k_u^+ c_u + \theta_u^+} \alpha_{i,u,d}^{\text{leave}}(k_u^+) dt \right] \\
 &= \frac{1}{c_d} \left[ (c_u - \theta_u^-) \alpha_{i,u,d}^{\text{leave}}(k_u^-) + \right. \\
 &\quad \left. c_u \sum_{i=1}^{k_u^+ - k_u^- - 1} \alpha_{i,u,d}^{\text{leave}}(k_u^- + i) + \theta_u^+ \alpha_{i,u,d}^{\text{leave}}(k_u^+) \right].
 \end{aligned}$$

Then, the synchronization function (4.63) can be rewritten into a linear equation of the form (6.32). Due to the definition of the control time interval, the synchronization formula will be the same in each control time interval. Taking the case in Fig. 6.1 for example, the synchronization functions within one control time interval are

$$\alpha_{i,u,d}^{\text{enter}}(k_d) = \alpha_{i,u,d}^{\text{leave}}(k_u), \quad (6.35)$$

$$\begin{aligned}
 \alpha_{i,u,d}^{\text{enter}}(k_d + 1) &= \frac{1}{c_d} \left[ (c_u - \Delta c_{u,d} - c_d) \alpha_{i,u,d}^{\text{leave}}(k_u) \right. \\
 &\quad \left. + (\Delta c_{u,d} + 2c_d - c_u) \alpha_{i,u,d}^{\text{leave}}(k_u + 1) \right], \quad (6.36)
 \end{aligned}$$

$$\begin{aligned}
 \alpha_{i,u,d}^{\text{enter}}(k_d + 2) &= \frac{1}{c_d} \left[ (2c_u - \Delta c_{u,d} - 2c_d) \alpha_{i,u,d}^{\text{leave}}(k_u + 1) \right. \\
 &\quad \left. + \Delta c_{u,d} \alpha_{i,u,d}^{\text{leave}}(k_u + 2) \right], \quad (6.37)
 \end{aligned}$$

where  $k_d = N_d k_c$  and  $k_u = N_c k_c$ . Therefore, the linear synchronization relationship can be pre-specified explicitly according to the given cycle times  $c_d$  and  $c_u$  of the corresponding intersections.

When the flow rate leaving link  $(u, d)$  is computed in the S model, the number of vehicles in downstream links  $n_{d,o}(k_o)$  is used to calculate the number of vehicles that the downstream links can accept at most. The simulation time step counter of intersection  $o$  is  $k_o$ . If  $k_o$  is different from  $k_d$ , an output synchronization function is needed for synchronizing the original number of vehicles in the downstream link of link  $(u, d)$ ,  $n_{d,o}^{\text{origin}}(k_o)$ , from time step  $k_o$  to  $k_d$ , as

$$n_{d,o}(k_d) = \mathcal{F}_{\text{out}}(n_{d,o}^{\text{origin}}(k_o), \dots, n_{d,o}^{\text{origin}}(k_o + \ell)), \quad (6.38)$$

which is also a linear expression that can be derived using the same rules as deriving the input synchronization function above.

### 6.3.3 Link time delay assumption

**Assumption 6.1** *We assume that the time delay of the vehicles traveling from the beginning of the link to the end of the queues in the link is constant over time and link.*

Then, having Assumption 6.1, (4.56) becomes linear as

$$\alpha_{u,d}^{\text{arriv}}(k_d) = (1 - \gamma_{\text{const}}) \cdot \alpha_{u,d}^{\text{enter}}(k_d - \tau_{\text{const}}) + \gamma_{\text{const}} \cdot \alpha_{u,d}^{\text{enter}}(k_d - \tau_{\text{const}} - 1), \quad (6.39)$$

where  $\tau_{\text{const}}$  and  $\gamma_{\text{const}}$  are constant values obtained by (4.57) with the queue length fixed. This queue length can be pre-calibrated for different traffic scenarios and environments according to the historical data, and stored in a database. Since the queue length in a link always changes over time, it is impossible to find an exact optimal constant queue length to suit the assumption. However, what we could do is to give the queue length ( $l$ ) several fixed values, e.g.  $l = 0$  when the traffic signal is green,  $l \neq 0$  when the traffic signal is red. The queue length for red signals can be scaled into several levels according to the traffic scenarios, and may also consider an environment factor. These queue length levels could be analyzed from the historical information of the link, and finally form a table that can be looked up. This method cannot derive very accurate time delays, but it can partially make up the errors for assuming the time delay constant.

With the reformulations above, the S model is reformulated into a mixed-integer linear model. Thereafter, an MILP method can be used to solve the optimization problem of the MPC controller based on the mixed-integer linear prediction model.

The reformulation above is not a standard and straightforward procedure that we could follow, it can be different for different characteristics of the problems we are focusing on. The main part that makes this work different from other's is how we tackle the synchronization between two successive intersections with different cycle times in Section 6.3.2. We linearized the synchronization equation, and gave a proof for it.



## 6.4 MILP-based MPC controller

For intersection  $d$ , the control time interval and the simulation time interval satisfy  $T_c = NN_d c_d$ . Then, for a given control time step  $k_c$ , the corresponding simulation time steps are  $k_d = NN_d k_c, NN_d k_c + 1, \dots, NN_d(k_c + 1) - 1$ .

After the model reformulation, the optimization problem of the MPC controller can be expressed as an MILP problem of the following form:

$$\begin{aligned}
 \min_{\mathbf{u}(k_c)} J_{\text{TTS}} &= \mathbf{c}^T \cdot \mathbf{u}(k_c) \\
 \text{s.t. } \mathbf{A} \mathbf{u}(k_c) &\leq \mathbf{b} \\
 \mathbf{A}_{\text{eq}} \mathbf{u}(k_c) &= \mathbf{b}_{\text{eq}} \\
 \mathbf{u}_{\min} &\leq \mathbf{u}(k_c) \leq \mathbf{u}_{\max} \\
 u_i(k_c) &\in \mathbb{Z} \text{ for } i \in \mathcal{B}
 \end{aligned} \tag{6.40}$$

for appropriately defined matrices  $\mathbf{A}$ ,  $\mathbf{A}_{\text{eq}}$ , and vectors  $\mathbf{c}$ ,  $\mathbf{b}$ ,  $\mathbf{b}_{\text{eq}}$ ,  $\mathbf{u}_{\min}$ ,  $\mathbf{u}_{\max}$ , and Set  $\mathcal{B}$ , where vector  $\mathbf{u}(k_c)$  contains all the optimization variables for control time steps  $k_c, \dots, k_c + N_p - 1$  (see A for more details).

The vector of optimized variables at control time step  $k_c$  in optimization problem (6.40) is

$$\begin{aligned}
 \mathbf{u}(k_c) &= [u^T(k_c|k_c) \ u^T(k_c + 1|k_c) \ \dots \\
 &\quad u^T(k_c + N_p - 1|k_c)]^T,
 \end{aligned} \tag{6.41}$$

where  $u(k_c + j|k_c)$  at any control time step consists of control variables (i.e. green time splits), state variables, and auxiliary variables for all the nodes and links in the traffic network as:

$$\begin{aligned}
 \mathbf{u}(\cdot) &= [ \overbrace{g^T(\cdot)}^{\text{Control variables}} \\
 &\quad \overbrace{q^T(\cdot) \ n^T(\cdot) \ n_{\text{downLink}}^T(\cdot) \ \alpha_{\text{leave}}^T(\cdot) \ \alpha_{\text{arriv}}^T(\cdot) \ \alpha_{\text{enter}}^T(\cdot)}^{\text{State variables}} \\
 &\quad \overbrace{\delta_1^T(\cdot) \ \delta_2^T(\cdot) \ f_1^T(\cdot) \ f_2^T(\cdot) \ z_1^T(\cdot) \ z_2^T(\cdot)}^{\text{Auxiliary variables}} ]^T.
 \end{aligned} \tag{6.42}$$

where  $(\cdot)$  stands for  $(k_c + j|k_c)$ ,  $n_{\text{downLink}}^T(k_c)$  represents the numbers of vehicles in the downstream links. All the optimized variables are real values except for the binary variables  $\delta_1(k_c)$  and  $\delta_2(k_c)$ . Supplied with initial traffic states and traffic demands of the network, the optimization problem can be solved at each control time step  $k_c$  by the MILP solver.

Several efficient branch-and-bound algorithms [46] are available for MILP problems. Moreover, there already exist several commercial and free solvers for MILP problems such as, e.g. CPLEX, Xpress-MP, GLPK, or lp\_solve (see [6, 89] for an overview).

## 6.5 S\* model-based MPC controller via MILP

### 6.5.1 S\* model

For the S model described in Chapter 4, the formula (4.53) computing the average flow rate leaving link  $(u, d)$  is the minimum of three terms. Each term gives the possible leaving flow rate under a traffic scenario. Under the saturated scenario, the average leaving flow rate depends on the saturated flow rate and the green time of the link; under the unsaturated scenario, the average flow rate is calculated according to the waiting and arriving flow rate at the intersection; under the over-saturated scenario, the average flow rate depends on the flow rate that the downstream link can accept. The traffic is always in the scenario that has the minimal average flow rate that could possible leave the link. As an urban traffic model, the S model is capable of describing all the situations that may happen in reality. However, when the S model is taken as a control model of the MPC controller, the third part of (4.53) can be removed from the S model to leave the over-saturated scenario out by adding extra constraints. Therefore, the S model can be rewritten into S\* model by rephrasing (4.53) by

$$\alpha_{u,d,o}^{\text{leave}}(k_d) = \min \left( \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_d) / c_d, q_{u,d,o}(k_d) / c_d + \alpha_{u,d,o}^{\text{arriv}}(k_d) \right), \quad (6.43)$$

and adding upper bound constraint  $0 \leq n_{u,d}(k_d) \leq C_{u,d}$  to traffic state  $n_{u,d}(k_d)$  (number of vehicles in a link) to make sure that the number of vehicles inside a link will not exceed its storage capacity  $C_{u,d}$ , i.e. no more vehicles can enter the link when it is already totally congested.

### 6.5.2 S\* model-based MPC controller

An MPC controller can be established based on the S\* model using the same method as shown in Section 6.4. A similar MILP optimization problem as (6.40) can be built through reformulating the S\* model into an MLD model. But, for the new MILP optimization problem, the number of the auxiliary variables is reduced by half because of the reduction of the S\* model. Although the S\* model does not take the over-saturated scenario into consideration, the free spaces of the downstream links can still be considered due to the constraints added. Instead of constraining the average traffic flow rates leaving links, the maximum number of vehicles that the downstream link can accept is then constrained by the upper bound. The traffic state  $n(k)$ , which is the number of vehicles in a link, is already an optimization variable of the MILP optimization problem. Hence, no extra effort is needed to add constraints to the traffic states  $n(k)$  of all the links within the network at every simulation time step  $k$ . In fact, the key idea of this approach lies in simplifying the optimization problem by reducing one equation in the prediction model (the S model), and adding upper bounds to the optimized state variables  $n(k)$  instead. As a result, the main complexity of the S\* model-based optimization problem (i.e. the number of auxiliary integer variables introduced) is reduced by half.

## 6.6 Case study

CORSIM is a microscopic traffic simulation software developed by FHWA [45], which can be used as a benchmark to design or test traffic control algorithms. We use CORSIM to

simulate the real traffic environment, and design MPC controllers according to Chapter 5 to decide control inputs for the traffic signals in CORSIM, as Fig. 6.2 shows. The on-line optimization of the MPC controller is reformulated into different optimization problems, which are solved using different optimization methods, and then the control performances (TTS) of the MPC controllers are compared. Multi-start Sequential Quadratic Programming (SQP) is applied to solve the original S model-based nonlinear non-convex optimization problem. An MILP solver is used to solve the S model-based or S\* model-based MILP problems obtained after reformulation according to Section 6.3.

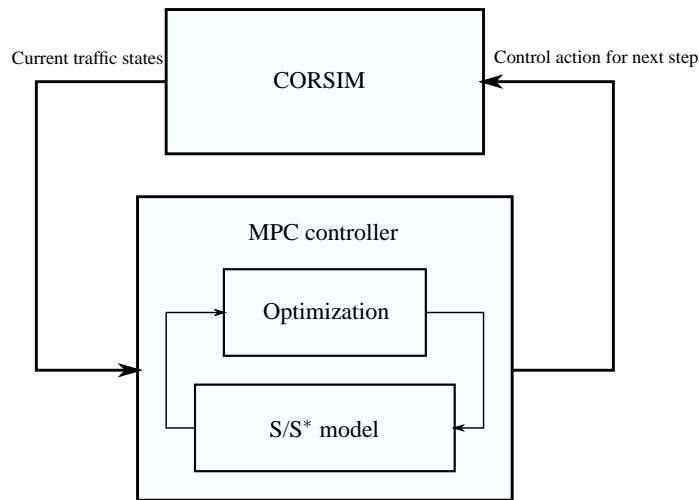


Figure 6.2: Illustration of the traffic control simulation

For the SQP solver, we apply `fmincon` provided by the optimization toolbox of Matlab. As MILP solver, we use CPLEX, implemented through the `cplex` interface function of the Matlab Tomlab toolbox.

The urban traffic network investigated is a grid network including 4 intersections (see Fig. 6.3). The cycle times are 120 s for intersection A and D, and 60 s for intersection B and C. The cycle times are constant, and off-sets are 0 during the simulation. The variation of cycle times and offsets is not considered in this case study, but we can distribute multiple control tasks to different layers of a hierarchical controller, so that the control task for a single layer is not too complex. Cycle times have comparatively tighter constraints, and do not have too much space for optimization. Therefore, we can adjust it on a higher level with a low frequency. Maybe we could try to add offsets as optimization variables in the lower level controller in the future. The control time interval is set to the least common multiple of all the cycle times in the network, i.e.  $T_c$  is 120 s. The prediction horizon is 10 control intervals. The control simulations run for the same time period of 3600s for all the experiments. The length of the links are 1220 m, all the links have 3 lanes. The average vehicle length is 5 m, and the free-flow speed on the links is 50 km/h. Therefore, the storage capacity of each link in the network is 732 veh, and the constant time delay of all the links is set to be 87.8 s. The input traffic flow rates to the network are constant. The simulations are carried out under 4 scenarios, according to different values of the input traffic flow rates

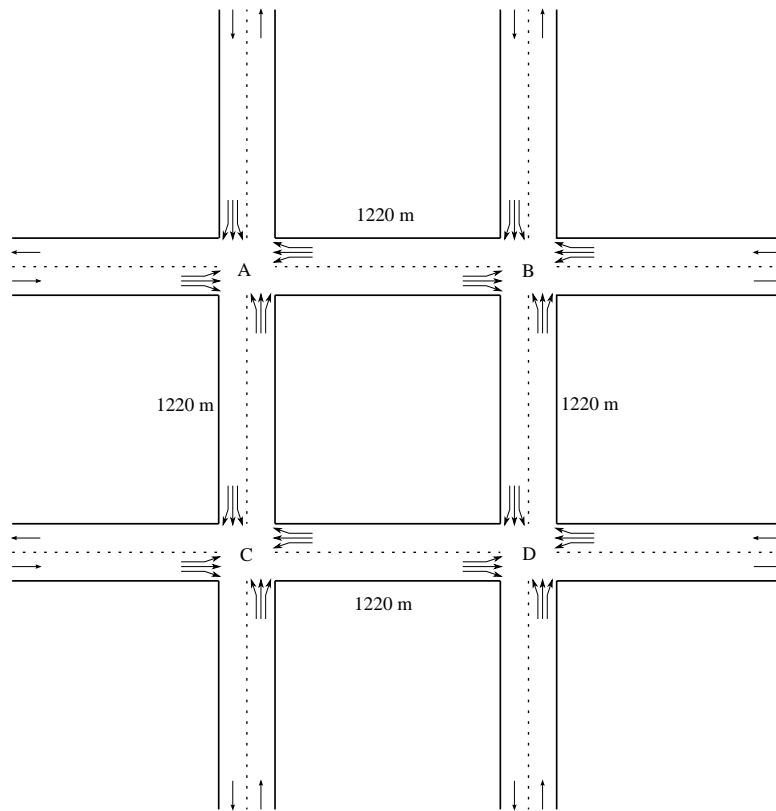


Figure 6.3: The layout of an urban road network

supplied to the network (input traffic demands), i.e. 500 veh/h, 1000 veh/h, 2000 veh/h, and 3000 veh/h. The simulation results are compared for these different traffic scenarios. The cost function is TTS for the entire simulation. The number of initial points for the SQP algorithm is 5. In the 5 initial points for SQP, one is the optimal solution derived in the optimization of the previous step (by shifting the optimal solution of the previous step one step forward, and adding a new decision to the end of the optimal solution sequence), the other 4 initial points are generated randomly within the lower and upper bounds.

MPC controllers are built for the urban traffic network based on different optimization algorithms. The MILP approaches for the reformulated S model and the reformulated S\* model are called respectively “S MILP” and “S\* MILP” here. The control performance (TTS) of the controllers at every control step since the beginning of the simulation is extracted from CORSIM, and compared in Fig. 6.4 to Fig. 6.7 for all the scenarios. In general, both S MILP and S\* MILP have either better performance (lower TTS) than, or equal performance to the nonlinear optimization algorithm, SQP. The reason is that the optimization problem at hand is a nonlinear non-convex problem because of the nonlinearity of the S model, so that it may have multiple local optima. The SQP algorithm is only able to search for the local optimum, which in general results in a sub-optimal solution. A multi-start method can be applied to help select a better sub-optimal solution. However, the multi-start

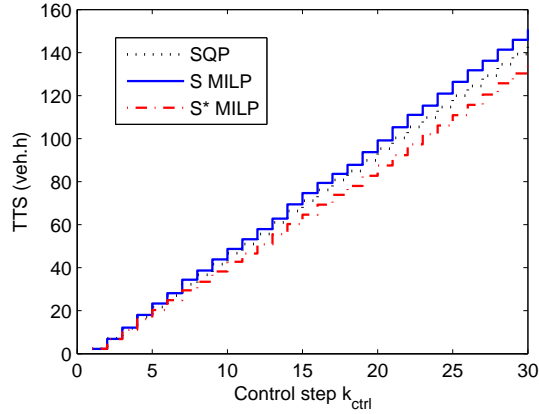


Figure 6.4: TTS comparison of the SQP, S MILP, and S\* MILP approaches for 500 veh/h

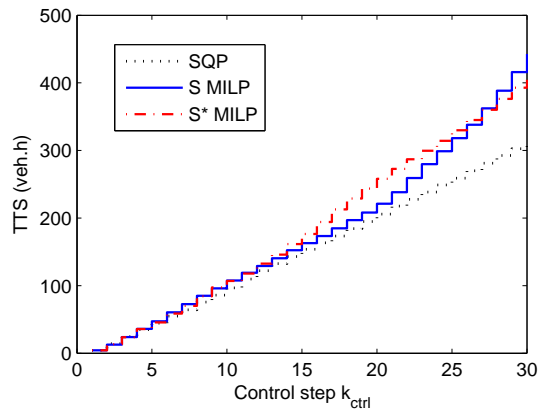


Figure 6.5: TTS comparison of the SQP, S MILP, and S\* MILP approaches for 1000 veh/h

procedure also results in more CPU time. On the contrary, an MILP problem can be solved efficiently by existing solvers that guarantee the global optimum.

We can see from Fig. 6.4 and Fig. 6.5 that the SQP algorithm has better performance (lower TTS) than S MILP, when the traffic flow demands are low. This is mainly caused by Assumption 6.1 made during the model reformulation. Recall that in order to turn the optimization problem into an MILP problem, Assumption 6.1 is made to linearize the original model. In the assumption, the time delay for vehicles running from the beginning of the link to the end of the queues in the link is considered to be constant. In the situation with high traffic demands, the number of leaving vehicles depends on the saturated flow rate of the link. In that case, the assumption almost does not have any influence on the results of MILP. However, in the situation with low traffic demands, the number of leaving vehicles from the link depends mainly on the number of waiting vehicles in the queues, which will be affected by the vehicles arriving from upstream after a certain time delay in the link.

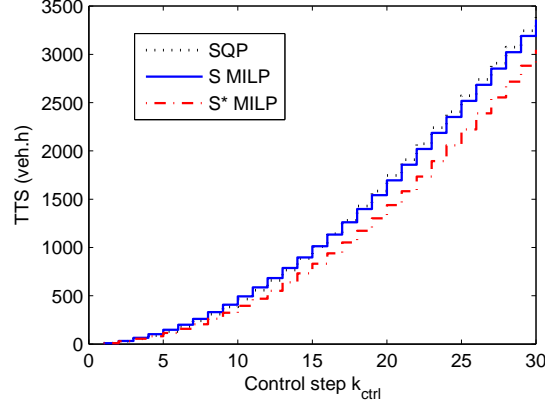


Figure 6.6: TTS comparison of the SQP, S MILP, and S\* MILP approaches for 2000 veh/h

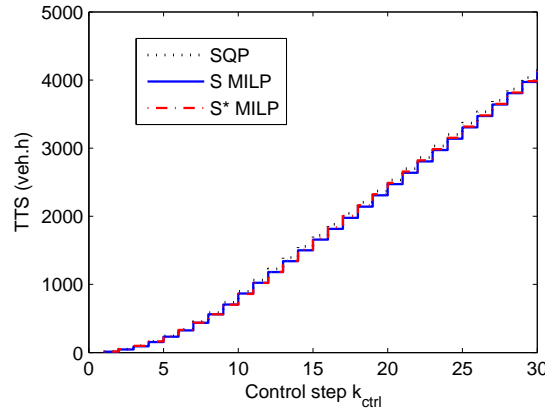


Figure 6.7: TTS comparison of the SQP, S MILP, and S\* MILP approaches for 3000 veh/h

Therefore, the assumption causes a mismatch between the original optimization problem and the reformulated MILP problem. As a result, the MILP algorithm fails to achieve better results than the SQP algorithm, when the network is less crowded (low traffic demands). The reduced S\* MILP is able to keep similar control performance as S MILP, and in some situations it performs even better. This is because MILP-S\* leaves out the third term (i.e. the supply of the receiving link) in the equation for calculating the departure traffic flow and thereby, it also implicitly introduces another constraint

$$\alpha_{u,d,o}^{\text{leave}}(k_d) = \min \left( \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_d) / c_d, q_{u,d,o}(k_d) / c_d + \alpha_{u,d,o}^{\text{arriv}}(k_d) \right) \leq C_{d,o} - n_{d,o}(k_d) \quad (6.44)$$

to the S\* MILP problem, which means that the demand of  $(u, d)$  cannot exceed the supply of the receiving link. This extra constraint has a function to inform the upstream links to control their permissions to the coming traffic demands, if a spillback may happen in the

downstream link. As a result, when the network is unsaturated, there is more space to allocate traffic demands more reasonably, thus MILP-S\* is able to achieve a better control performance.

When the traffic flow demands are high, and the traffic network is more crowded (saturated), the MILP approaches achieve better performance than the SQP approach, as Fig. 6.6 to Fig. 6.7 shows. The influence of Assumption 6.1, as in low demand scenarios, almost disappears then. But, due to the high traffic demands and traffic density, there is also less space for the MILP approaches to improve the control performance, and hence, the TTS curves stay very close (see Fig. 6.7).

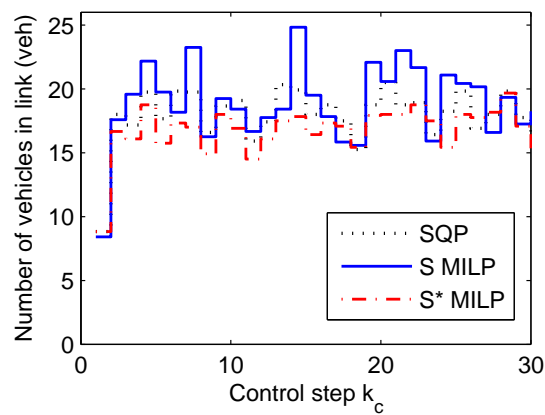


Figure 6.8: The evolution of the average number of vehicles in link (A,B) when network traffic flow demands are 500 veh/h

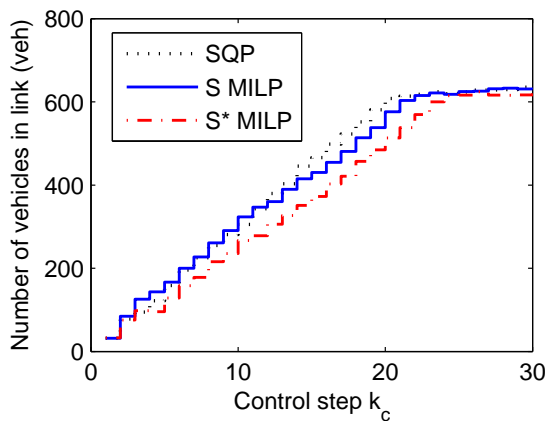


Figure 6.9: The evolution of the average number of vehicles in link (A,B) when network traffic flow demands are 2000 veh/h

In Fig. 6.8 and Fig. 6.9, we give the evolution of the average number of vehicles in link (A,B) for two scenarios with network traffic flow demands of 500 veh/h and 2000 veh/h respectively. We can see from the figures that the number of vehicles in link (A,B) fluctuates over time in the scenario with 500 veh/h, while the number of vehicles in link (A,B) accumulates until the link becomes saturated in the scenario with 2000 veh/h, which illustrates that the vehicles spill back to the upstream links.

Table 6.2: Comparison of computation times and the number of optimization variables for different optimization algorithms in all scenarios

Scenario	Algorithm	CPU time (s)		# variables	
		$t_{\text{avg}}$	$t_{\text{max}}$	Real	Boolean
500 veh/h	SQP	461.4	601.7	120	-
	S MILP	0.8	2.9	6880	1440
	S* MILP	1.1	1.2	4480	720
1000 veh/h	SQP	459.1	548.5	120	-
	S MILP	1.3	2.5	6880	1440
	S* MILP	1.3	1.9	4480	720
2000 veh/h	SQP	453.4	552.5	120	-
	S MILP	1.2	2.3	6880	1440
	S* MILP	1.6	2.5	4480	720
3000 veh/h	SQP	452.4	526.4	120	-
	S MILP	1.1	2.6	6880	1440
	S* MILP	1.1	1.5	4480	720

In Table 6.2, the computation time and the number of optimization variables are compared for the different optimization approaches, where “ $t_{\text{avg}}$ ” is the average optimization CPU time over all the control steps, and “ $t_{\text{max}}$ ” is the maximum optimization CPU time. The SQP approach does not have boolean optimization variables. S\* MILP has less optimization variables than S MILP, where the number of auxiliary variables is reduced by half because of the model adaptation. In general, the MILP problem with less boolean variables will be solved faster than the one with more boolean variables, due to the branch-and-search procedure of MILP solvers. But, this is not always true for the simulation results of S MILP and S\* MILP. Nevertheless, S MILP and S\* MILP problems can be both solved very fast by MILP solvers. The CPU times are reduced significantly from hundreds of seconds to a few seconds compared to the SQP solver. The number of initial points selected for SQP is 5. Therefore, the average CPU time for one single run of SQP can be computed by dividing 5. Thus, taking the scenario with 500 veh/h as an example, the average CPU time for a single run of SQP is 92.3 s, the average CPU time for S MILP is 0.8 s, and the average CPU time for S\* MILP is 1.1 s. Therefore, by reformulating the original nonlinear non-convex optimization problem into an MILP problem, the MPC controller for urban traffic network becomes much more time efficient on-line.



## 6.7 Conclusion

Model predictive control provides many advantages for controlling urban traffic networks. But it also has a high requirement for the computational efficiency of the on-line optimization. Due to the nonlinear non-convex nature of the optimization problem, the on-line computational complexity is a big challenge for the MPC controller. To solve this problem, in this chapter, the nonlinear  $S$  model was reformulated into a model, which can be expressed by mixed-integer linear equalities and inequalities. The  $S$  model and the reduced  $S^*$  model are both reformulated according to this method, and the original nonlinear non-convex optimization problem is written in the form of MILP problems based on the reformulated  $S$  model and  $S^*$  model respectively. An efficient MILP solver can then be applied to solve the reformulated MILP optimization problems of MPC.

The simulation experiments indicate that the MILP-based approaches may maintain the same control performance as the multi-start SQP-based control approach, and sometimes can achieve even better control performance. However, in the situation of low traffic demands and traffic density, the assumption made during the model reformulation may cause a mismatch between the reformulated MILP problem with the original optimization problem. However, this mismatch can be alleviated by calibrating the link time delay beforehand for the low traffic demand scenario. The biggest advantage of the MILP-based MPC controllers is that the on-line computational speed is increased dramatically compared to the original MPC controller (e.g. in the case study, the time for solving the optimization problem is reduced from hundreds of seconds to only a few seconds). This indicates that the MILP approach is a potential method that can be selected to reduce the on-line computational complexity of the  $S$  model-based MPC controller, and to further increase the applicability of the MPC controller in real-life traffic networks.



## Chapter 7

# Integrated MPC for the Reduction of Travel Delays and Emissions

In urban areas, the density of the population is relatively high. People living in big cities usually live busy lives, and suffer comparatively worse environmental conditions (less living space, more air pollution, etc.). One of the biggest sources of the environmental pollution in cities comes from the emissions of the busy traffic flows. A well-designed urban traffic management system, which can control both travel delays and traffic emissions effectively, and accordingly make the transportation more efficient and comfortable, is very important. Therefore, integrated urban traffic control strategies aiming at reducing both travel delays and emissions will be discussed in this chapter<sup>1</sup>.

### 7.1 Introduction

The emissions of vehicles contain several harmful substances, such as nitrogen oxides ( $\text{NO}_x$ , such as nitrogen monoxide, nitrogen dioxide), hydrocarbons (HC), carbon monoxide (CO), carbon dioxide ( $\text{CO}_2$ ), and fine particulate matter.  $\text{NO}_x$  may participate in several reactions after being released into the open air, and thus generate ozone, acid rain, and fine particles. Among them, the generated ground-level ozone may trigger reactions in people who have asthma, acid rain will cause damage to soil, agriculture, water, etc. Fine particles will give rise to sufferance from respiratory or cardiovascular diseases. HC, such as methane, ethane, propane, irritates the mucous membranes, and causes headaches, liver damage, and even cancer. CO is a colorless and odorless gas generated by incomplete combustion of gasoline. Driving is the cause of over half of global CO emissions. CO can decrease the ability of the blood to carry oxygen, and may cause heart disease.  $\text{CO}_2$  is not directly harmful to humans, but is very bad for the environment. Because it is a very important gas that causes the greenhouse effect. The gases released by vehicles make up 14% of the total green house

---

<sup>1</sup>The content of this chapter was published in [86].

gases released around the world. Due to the high concentration of transportation, the vehicle emissions will form big smog over the city areas, and thus deteriorate the climate of large cities. In general, traffic pollution deteriorates our living environment, thus increases the risk for the people who already have heart or lung diseases. Therefore, it is very necessary to integrate traffic emissions control into the urban traffic management system, so as to provide a healthier, safer, and more comfort living environment for the people living in urban areas.

So far, most of the ongoing research is focusing on reducing traffic delays and traffic congestion, and improving the traffic flow throughput. However, in some circumstances, an increased traffic flow throughput may result in even higher total traffic emissions [129]. In general, we cannot take for granted that the smaller travel delay is, the less traffic emissions will be generated. In fact, the emissions of a vehicle depend greatly on the operational conditions of the vehicle [4, 28, 29]. Large emissions can be given out by a vehicle with either too high speed or too low speed. Therefore, an integrated control strategy is necessary that balances performance in terms of both travel delays and all types of traffic emissions. Traffic control strategies considering both travel delays and traffic emissions for highways have been already discussed in [129–131]. Since the speed range and the behavior of vehicles are different for urban and highway, in this chapter we will address travel delays and traffic emissions for urban areas.

To this aim, first a macroscopic urban traffic model that also estimates the emissions of traffic flows is proposed, and an MPC controller considering both traffic delays and emissions is built using this model as the prediction model. Since at each step the MPC controller solves an optimization problem on-line, it has high requirements for the on-line computational complexity of the prediction model. The S model in Chapter 4 is taken as the prediction model for the MPC controller. This model is a macroscopic urban traffic model, which is fast to compute and also accurate enough for control purposes [82]. In order to well capture the emissions of a vehicle running on a road in urban area, a microscopic traffic emission model that is based on both velocity and acceleration, is selected. This vehicle emission model provides reasonable estimates, when the vehicle is decelerating, accelerating, or moving slowly in front of the stop-line in red signals. Integrated with this microscopic traffic emission model, the overall macroscopic prediction model is able to provide estimations of both travel delays and emissions for the MPC controller.

## 7.2 Microscopic traffic emission and fuel consumption model

Vehicle emissions depend on many factors, such as vehicle status (like engine, chassis, age, and maintenance), environmental conditions (such as infrastructure and weather), and operational factors (such as speed, acceleration, and engine load). These last factors are the most decisive elements for the fuel consumption and the emission of harmful substances. A traffic emission and fuel consumption model calculates the quantity of the generated emissions and consumed fuel based on the operating conditions of the vehicles.

Technology-based emission or fuel consumption models are very detailed models. These kind of models are developed for a specific vehicle (or engine) model [62]. Such models are used for the assessment of new technological developments, and for regulation purposes [62]. Since these models are very detailed, they are difficult to use for online prediction or

on-line estimation of emissions and fuel consumption of traffic flow. Therefore, for computational reasons it is advisable not to use such models for on-line model-based traffic control purposes [129].

There are simpler emission and fuel consumption models that are suitable for control purposes, i.e. average-speed-based models and dynamic-based models.

Average-speed-based models calculate the emissions and fuel consumption of each vehicle based on the average traveling speed of the vehicle. This average traveling speed can be calculated either over the entire trip, or over some local time periods to take some variations of the speed into consideration [19]. The mathematical equations for CO, NO<sub>x</sub>, and HC emissions from this model are:

$$E_{\text{CO}}(\bar{v}) = (0.001728\bar{v}^2 - 0.245\bar{v} + 9.617) \text{ [g/km]} \quad (7.1)$$

$$E_{\text{NO}_x}(\bar{v}) = 10^{-4}(0.854\bar{v}^2 - 85\bar{v} + 5260) \text{ [g/km]} \quad (7.2)$$

$$E_{\text{HC}}(\bar{v}) = 10^{-4}(0.521\bar{v}^2 - 88.8\bar{v} + 4494) \text{ [g/km]} \quad (7.3)$$

where  $\bar{v}$  is the average speed of a vehicle on a given route, and  $E_{\text{CO}}$ ,  $E_{\text{NO}_x}$ , and  $E_{\text{HC}}$  are emission levels of carbon monoxide, nitrogen oxides and hydrocarbons in g/km. This model is only based on the average speed of a vehicle.

However, dynamic-based models use more detailed knowledge of the vehicle dynamics, i.e. the speed and acceleration data of each vehicle at every time instant. As they are microscopic traffic emission and fuel consumption models, dynamic-based models are more accurate than the average-speed-based models.

VT-micro [4] is a microscopic dynamic-based traffic emission and fuel consumption model. It evaluates the emissions based on not only the speed of every vehicle, but also the acceleration or the deceleration of each vehicle. VT-micro generates emissions and fuel consumption of an individual vehicle with index  $i$  at every time step  $k$  based on the current speed  $v_i(k)$  and acceleration  $a_i(k)$  of the vehicle, as

$$E_{\theta,i}(v_i(k), a_i(k)) = \exp(\tilde{\mathbf{v}}_i^T(k) \mathbf{P}_\theta \tilde{\mathbf{a}}_i(k)), \quad (7.4)$$

where  $E_{\theta,i}$  stands for the emission or fuel consumption for  $\theta \in \tilde{\mathcal{M}} = \{\text{CO}, \text{NO}_x, \text{HC}, \text{FC}\}$ , and the vectors of velocities and accelerations with the exponents going up from 0 to 3 are defined as  $\tilde{\mathbf{v}}_i(k) = [1 \ v_i(k) \ v_i^2(k) \ v_i^3(k)]^T$ ,  $\tilde{\mathbf{a}}_i(k) = [1 \ a_i(k) \ a_i^2(k) \ a_i^3(k)]^T$ , while  $\mathbf{P}_\theta$  is the parameter matrix of the model for emission or fuel consumption type  $\theta \in \tilde{\mathcal{M}}$ . The matrices  $\mathbf{P}_\theta$  for the emission variables  $\theta = \{\text{CO}, \text{HC}, \text{NO}_x\}$  and the fuel consumption are initially given in [4, 5], and are adapted for metric system as:

$$P_{\text{CO}} = 10^{-2} \begin{bmatrix} 88.7447 & 48.8324 & 32.8837 & -4.7675 \\ 23.2920 & 4.1656 & -3.2843 & 0 \\ -0.8503 & 0.3291 & 0.5700 & -0.0532 \\ 0.0163 & -0.0082 & -0.0118 & 0 \end{bmatrix}, \quad (7.5)$$

$$P_{\text{HC}} = 10^{-2} \begin{bmatrix} -72.8040 & 0 & 25.1563 & -0.3284 \\ 8.1857 & 10.9200 & -1.9423 & -1.2745 \\ -0.2260 & -0.3531 & 0.4356 & 0.1258 \\ 0.0069 & 0.0072 & -0.0080 & -0.0021 \end{bmatrix}, \quad (7.6)$$

$$P_{\text{NO}_x} = 10^{-2} \begin{bmatrix} -106.7680 & 83.4524 & 9.5433 & -3.3549 \\ 15.2306 & 16.6647 & 10.1565 & -3.7076 \\ -0.1830 & -0.4591 & -0.6836 & 0.0737 \\ 0.0020 & 0.0038 & 0.0091 & -0.0016 \end{bmatrix}, \quad (7.7)$$

and

$$P_{\text{fuel}} = 10^{-2} \begin{bmatrix} -67.9440 & 44.3809 & 17.1641 & -4.2024 \\ 9.7326 & 5.1753 & 0.2942 & -0.7068 \\ -0.3014 & -0.0742 & 0.0109 & 0.0116 \\ 0.0053 & 0.0006 & -0.0010 & -0.0006 \end{bmatrix}. \quad (7.8)$$

The emissions of  $\text{CO}_2$  are proportional to vehicle fuel consumption, i.e. the more fuel consumed, the more  $\text{CO}_2$  will be released. Therefore, the emissions of  $\text{CO}_2$  can be derived from the fuel consumption model as in [130]:

$$E_{\text{CO}_2,i}(v_i(k), a_i(k)) = 1.17 \times 10^{-5} + 2.65 E_{\text{Fuel},i}(v_i(k), a_i(k)) \text{ [l/km]}. \quad (7.9)$$

Fig. 7.1 illustrates the variations of the emissions for VT-micro model as a function of the speed of the vehicles.

In this chapter, we are going to use VT-micro model to integrate with the S model so as to estimate the vehicle emissions and fuel consumption for urban traffic.

## 7.3 Integrated traffic flow, traffic emission, and fuel consumption model

### 7.3.1 Urban traffic behaviors for individual vehicles

As a microscopic model, the VT-micro model provides the emissions of an individual vehicle at a certain location and a time instant. But, as a macroscopic model, the S model only provides information of traffic flows instead of every detail of each individual vehicle. However, the S model can capture the main behavior of the vehicles, when they are running along a road. The time period spent by a vehicle running along a road can be divided into several parts, in each of which the behavior of the vehicle is assumed to be uniform. Define the set of the behaviors as  $B = \{\text{free, idling, dec, acc, nonstop}\}$ . Fig. 7.2 shows how the velocity of a vehicle could vary in different behavior regions, when it travels along an urban road. Therefore, the emission and fuel consumption model for an individual vehicle in behavior  $b \in B$  can be derived for  $\theta \in M = \{\text{CO, CO}_2, \text{NO}_x, \text{HC, FC}\}$  according to Section 7.2.

As Fig. 7.2(a) shows, in the regions “free” and “idling”, the vehicle runs with a constant velocity, i.e. the acceleration is  $a = 0$ . The region “free” stands for the time period that the vehicle is running on the link with free-flow speed  $v = v_{\text{free}}$ , while the region “idling” represents the time period that the vehicle is moving in a queue in front of an intersection with a very low speed  $v = v_{\text{low}}$ . Therefore, the emission functions for the vehicle running with free-flow speed and the vehicle idling with a very low speed in the queues are respectively

$$E_{\theta,i}^{\text{free}}(k) = E_{\theta,i}(v_{\text{free}}, 0), \quad (7.10)$$

$$E_{\theta,i}^{\text{idling}}(k) = E_{\theta,i}(v_{\text{low}}, 0). \quad (7.11)$$

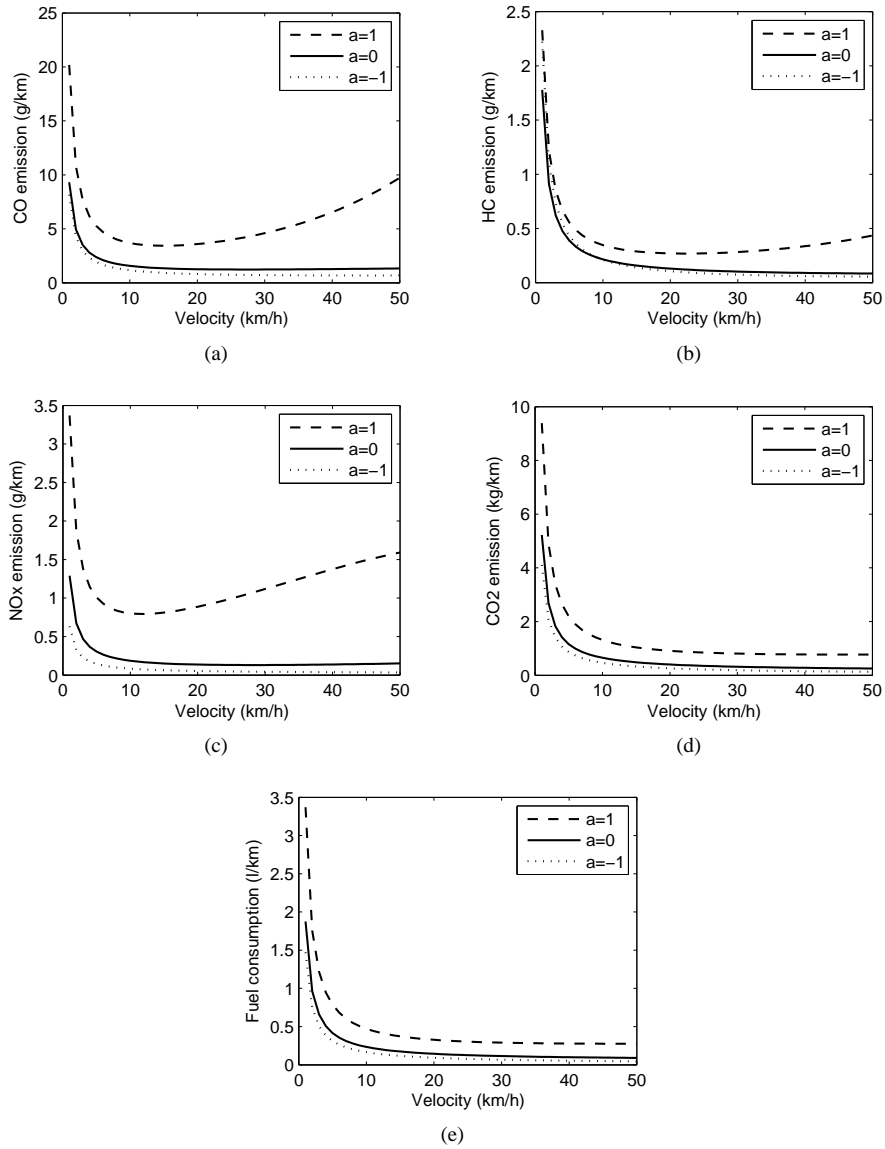


Figure 7.1: Vehicle emissions of VT-micro for CO, HC, and NO<sub>x</sub>, CO<sub>2</sub>, and fuel consumption at various acceleration  $a$  ( $m/s^2$ )

The regions “dec” and “acc” respectively represent the deceleration and acceleration behavior of the vehicle near an intersection. Here, the assumption is made that the vehicle will decelerate and accelerate with constant acceleration  $a_{dec} < 0$  and  $a_{acc} > 0$  respectively. The average velocity  $v_{avg}$  is used in the emission function to approximate the velocity during decelerating and accelerating. Then, the emission functions for the vehicle decelerating and

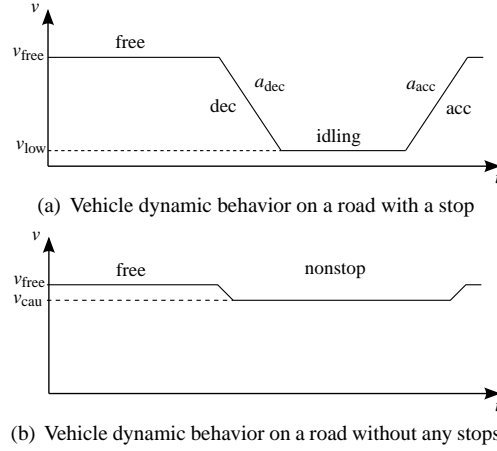


Figure 7.2: Vehicle dynamic behavior on a road

accelerating are

$$E_{\theta,i}^{\text{dec}}(k) = E_{\theta,i}(v_{\text{avrg}}, a_{\text{dec}}), \quad (7.12)$$

$$E_{\theta,i}^{\text{acc}}(k) = E_{\theta,i}(v_{\text{avrg}}, a_{\text{acc}}), \quad (7.13)$$

where the average velocity of the vehicle is an average of the velocity before and after acceleration:  $v_{\text{avrg}} = (v_{\text{free}} + v_{\text{low}})/2$ .

If the vehicle arrives at the stop line, where no queue is in front of it and the traffic light is also green, then the vehicle will leave the link without a stop at a constant speed. This constant speed,  $v_{\text{cau}}$ , is a little bit lower than the free-flow speed, because drivers will in general be more cautious while passing the intersections. Therefore, the emissions for the nonstop vehicles are

$$E_{\theta,i}^{\text{nonstop}}(k) = E_{\theta,i}(v_{\text{cau}}, 0). \quad (7.14)$$

**Remark 7.1** In this subsection, all the variables are assumed to be the same for a vehicle on any link. If the locations of vehicles are considered, then the emission  $E_{\theta,i}^b$  of vehicle  $i$  on link  $(u, d)$  in behavior  $b$  should be remarked as  $E_{\theta,u,d,i}^b$ .  $\square$

### 7.3.2 Integrated VT-S traffic emission and fuel consumption model

The S model provides macroscopic traffic states for each link  $(u, d) \in L$  in each simulation time interval (cycle time). The traffic states include the number of vehicles traveling with free-flow speed, the number of vehicles decelerating and accelerating, the number of vehicle waiting in queues. Based on this macroscopic information and the microscopic emission and fuel consumption model of the previous section, a macroscopic traffic emission and fuel consumption model can be obtained by combining the macroscopic S model and the VT-micro model, which results in a macroscopic integrated traffic flow, emission, and fuel consumption model, which we call the VT-S model.



The VT-S model for emission and fuel consumption  $\theta \in M$  in link  $(u, d) \in L$  during time period  $[c_d \cdot k_d, c_d \cdot (k_d + 1)]$  is

$$\begin{aligned} E_{\theta, u, d}(k_d) &= \sum_{b \in B} \sum_{k \in \mathcal{X}(d, k_d)} T \cdot \sum_{i \in \mathcal{V}(b, u, d, k)} E_{\theta, u, d, i}^b(k) \\ &= \sum_{b \in B} E_{\theta, u, d}^b(k_d) \cdot N_{u, d}^b(k_d) \cdot t_{u, d}^b(k_d), \end{aligned} \quad (7.15)$$

where  $\mathcal{V}(b, u, d, k)$  is the set of vehicles that have behavior  $b$  at time step  $k$  in link  $(u, d)$ ,  $\mathcal{X}(d, k_d)$  is the set of time steps  $k$  such that  $kT \in [c_d \cdot k_d, c_d \cdot (k_d + 1)]$  at which the vehicles are in behavior  $b$  in link  $(u, d)$ ,  $E_{\theta, u, d}^b(k_d)$  is the constant traffic emission for emission  $\theta$  of a vehicle on link  $(u, d)$  with behavior  $b$  during time period  $[c_d \cdot k_d, c_d \cdot (k_d + 1)]$ ,  $N_{u, d}^b(k_d)$  is the number of vehicles that have behavior  $b$  in link  $(u, d)$  during time period  $[c_d \cdot k_d, c_d \cdot (k_d + 1)]$ , and  $t_{u, d}^b(k_d)$  is the time period that the vehicles keep having this behavior.

Urban traffic states on a link can be separated into different scenarios according to the level of the traffic density. In the saturated traffic scenario, the queues of vehicles resulting from the red phase cannot be dissolved completely at the following green phase, i.e. all the arriving vehicles have to stop and wait once for the next green light to leave the link. For the over-saturated traffic scenario, the vehicles need to wait for even more cycle times in the queues than in a saturated scenario. On the contrary, in the unsaturated traffic scenario, all the accumulated vehicles during the red phase are able to leave the link in the following green phase, and some vehicles can even leave the link without any stop. Since the traffic behaviors could differ between these scenarios, the VT-S model can be further derived for each of the three scenarios.

First, we are going to recall some of the notations of the S model that will be used in the following content:

$O_{u, d}$	: set of downstream nodes of output links of link $(u, d)$ ,
$n_{u, d}(k_d)$	: number of vehicles in link $(u, d)$ at step $k_d$ ,
$q_{u, d}(k_d)$	: queue length (expressed as the number of vehicles) at step $k_d$ in link $(u, d)$ ,
$\alpha_{u, d}^{\text{arriv}}(k_d)$	: average flow rate arriving at the tail of the queue in link $(u, d)$ at step $k_d$ ,
$\beta_{u, d, o}(k_d)$	: fraction of the traffic in link $(u, d)$ anticipating to turn to link $(d, o)$ at step $k_d$ ,
$\mu_{u, d}$	: saturation flow rate leaving link $(u, d)$ ,
$g_{u, d, o}(k_d)$	: green time length during step $k_d$ for the traffic stream towards link $(d, o)$ in link $(u, d)$

### Saturated scenario

In the saturated scenario, not all the vehicles waiting and arriving in the queues could leave the link in the current green phase. So some vehicles have to wait until the next green phase, i.e. the number of vehicles waiting and arriving in the link exceeds the maximum number of vehicles that could leave at most in one cycle time, however the queues waiting in the link can be dissolved in the following green phase. This is characterized by the following

condition:

$$\begin{aligned} q_{u,d}(k_d) &\leq \sum_{o \in O_{u,d}} \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_d) \\ &\leq c_d \cdot \alpha_{u,d}^{\text{arriv}}(k_d) + q_{u,d}(k_d). \end{aligned} \quad (7.16)$$

Therefore, all the vehicles have to wait once for a red traffic signal in the queues before leaving the link, i.e. no vehicle can leave the link without stop. For the saturated scenario, the number of vehicles that have behavior  $b \in B$  in link  $(u, d)$  during time period  $[c_d \cdot k_d, c_d \cdot (k_d + 1)]$  is given by

$$N_{u,d}^{\text{free}}(k_d) = n_{u,d}(k_d) - c_d \cdot \alpha_{u,d}^{\text{arriv}}(k_d) - q_{u,d}(k_d) \quad (7.17)$$

$$N_{u,d}^{\text{idling},1}(k_d) = c_d \cdot \alpha_{u,d}^{\text{arriv}}(k_d) + q_{u,d}(k_d) - \quad (7.18)$$

$$\sum_{o \in O_{u,d}} \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_d) \quad (7.19)$$

$$N_{u,d}^{\text{idling},2}(k_d) = \sum_{o \in O_{u,d}} \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_d) - q_{u,d}(k_d) \quad (7.20)$$

$$N_{u,d}^{\text{idling},3}(k_d) = 0 \quad (7.21)$$

$$N_{u,d}^{\text{idling},4}(k_d) = q_{u,d}(k_d) \quad (7.22)$$

$$N_{u,d}^{\text{dec}}(k_d) = c_d \cdot \alpha_{u,d}^{\text{arriv}}(k_d) \quad (7.23)$$

$$N_{u,d}^{\text{acc}}(k_d) = \sum_{o \in O_{u,d}} \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_d) \quad (7.24)$$

$$N_{u,d}^{\text{nonstop}}(k_d) = 0, \quad (7.25)$$

and the time periods that the vehicles keep having this behavior in link  $(u, d)$  during time period  $[c_d \cdot k_d, c_d \cdot (k_d + 1)]$  are given by

$$t_{u,d}^{\text{free}}(k_d) = c_d \quad (7.26)$$

$$t_{u,d}^{\text{idling},1}(k_d) = c_d - (v_{\text{low}} - v_{\text{free}})/a_{\text{dec}} \quad (7.27)$$

$$t_{u,d}^{\text{idling},2}(k_d) = c_d - (v_{\text{low}} - v_{\text{free}})/a_{\text{dec}} - (v_{\text{free}} - v_{\text{low}})/a_{\text{acc}} \quad (7.28)$$

$$t_{u,d}^{\text{idling},3}(k_d) = 0 \quad (7.29)$$

$$t_{u,d}^{\text{idling},4}(k_d) = c_d - (v_{\text{free}} - v_{\text{low}})/a_{\text{acc}} \quad (7.30)$$

$$t_{u,d}^{\text{dec}}(k_d) = (v_{\text{low}} - v_{\text{free}})/a_{\text{dec}} \quad (7.31)$$

$$t_{u,d}^{\text{acc}}(k_d) = (v_{\text{free}} - v_{\text{low}})/a_{\text{acc}} \quad (7.32)$$

$$t_{u,d}^{\text{nonstop}}(k_d) = 0. \quad (7.33)$$

Equation (7.17) gives the number of vehicles that are running on link  $(u, d)$  with free-flow speed during the time period shown in (7.26). The vehicles idling in front of the stop-line in link  $(u, d)$  can be classified into four groups:

1. Vehicles idling for the rest of the cycle time after deceleration;

2. Vehicles idling between deceleration and acceleration;
3. Vehicles idling for the entire cycle time;
4. Vehicles idling for the rest of the cycle time before acceleration.

In the saturated scenario, (7.19) gives the number of vehicles that arrive at the end of the queues and decelerate to a low speed in link  $(u, d)$ , and then keep idling for the time period as in (7.27). Equation (7.20) gives the number of vehicles that decelerate to arrive at the end of the queues, keep idling for time period in (7.28), and then accelerate to leave link  $(u, d)$ . Equation (7.22) gives the number of vehicles in the queues that keep idling for the time period as in (7.30), and finally accelerate and leave link  $(u, d)$ . All the vehicles arriving at the end of the queues need to decelerate as (7.23) shows, and all the vehicles leaving link  $(u, d)$  will accelerate as (7.24) shows.

### Over-saturated scenario

In the over-saturated scenario, the vehicles waiting in the queues could not leave the link in the current green phase. So, some vehicles have to wait more than two red traffic phases, i.e. the number of vehicles waiting in the queues to leave the link exceeds the maximum number of vehicles that could leave at most in one cycle time:

$$\sum_{o \in O_{u,d}} \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_d) < q_{u,d}(k_d). \quad (7.34)$$

For the over-saturated scenario, the number of vehicles that have behavior  $b \in B$  in link  $(u, d)$  during time period  $[c_d \cdot k_d, c_d \cdot (k_d + 1)]$  is given by

$$N_{u,d}^{\text{free}}(k_d) = n_{u,d}(k_d) - c_d \cdot \alpha_{u,d}^{\text{arriv}}(k_d) - q_{u,d}(k_d) \quad (7.35)$$

$$N_{u,d}^{\text{idling},1}(k_d) = c_d \cdot \alpha_{u,d}^{\text{arriv}}(k_d) \quad (7.36)$$

$$N_{u,d}^{\text{idling},2}(k_d) = 0 \quad (7.37)$$

$$N_{u,d}^{\text{idling},3}(k_d) = q_{u,d}(k_d) - \sum_{o \in O_{u,d}} \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_d) \quad (7.38)$$

$$N_{u,d}^{\text{idling},4}(k_d) = \sum_{o \in O_{u,d}} \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_d) \quad (7.39)$$

$$N_{u,d}^{\text{dec}}(k_d) = c_d \cdot \alpha_{u,d}^{\text{arriv}}(k_d) \quad (7.40)$$

$$N_{u,d}^{\text{acc}}(k_d) = \sum_{o \in O_{u,d}} \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_d) \quad (7.41)$$

$$N_{u,d}^{\text{nonstop}}(k_d) = 0, \quad (7.42)$$

and the time periods that the vehicles keep having this behavior in link  $(u, d)$  during time period  $[c_d \cdot k_d, c_d \cdot (k_d + 1)]$  are given by

$$t_{u,d}^{\text{free}}(k_d) = c_d \quad (7.43)$$

$$t_{u,d}^{\text{idling},1}(k_d) = c_d - (v_{\text{low}} - v_{\text{free}})/a_{\text{dec}} \quad (7.44)$$

$$t_{u,d}^{\text{idling},2}(k_d) = 0 \quad (7.45)$$

$$t_{u,d}^{\text{idling},3}(k_d) = c_d \quad (7.46)$$

$$t_{u,d}^{\text{idling},4}(k_d) = c_d - (v_{\text{free}} - v_{\text{low}})/a_{\text{acc}} \quad (7.47)$$

$$t_{u,d}^{\text{dec}}(k_d) = (v_{\text{low}} - v_{\text{free}})/a_{\text{dec}} \quad (7.48)$$

$$t_{u,d}^{\text{acc}}(k_d) = (v_{\text{free}} - v_{\text{low}})/a_{\text{acc}} \quad (7.49)$$

$$t_{u,d}^{\text{nonstop}}(k_d) = 0. \quad (7.50)$$

Except for the “idling” behavior, all the above formulas are the same as in the saturated scenario. All the vehicles arriving at the end of the queues as shown in (7.36) will decelerate and be idling for time period (7.44). A part of the vehicles waiting in the queues as in (7.38) cannot leave link  $(u, d)$ , and will be idling for the entire cycle time. All the vehicles as shown in (7.39) will be idling for time period (7.47), and then accelerate and leave link  $(u, d)$ .

### Unsaturated scenario

In the unsaturated scenario, the queues can be dissolved before the current green phase ends. Thus, the traffic demand, i.e. the number of vehicles waiting and arriving to leave the link is less than the maximum number of vehicles that can leave in one cycle, which is characterized as

$$c_d \cdot \alpha_{u,d}^{\text{arriv}}(k_d) + q_{u,d}(k_d) < \sum_{o \in O_{u,d}} \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_d). \quad (7.51)$$

Therefore, during a green phase, the vehicles waiting in the queues can be considered to first leave the link according to the saturated flow rate of the link  $\mu_{u,d}$ , and then, after the queues are dissolved, the arriving vehicles will leave the link without a stop according to the arriving flow rate  $\alpha_{u,d}^{\text{arriv}}(k_d)$  in the rest of the green time. Hereafter, the green time for link  $(u, d)$  in the  $k_d$ th cycle,  $g_{u,d}(k_d)$ , can be approximately separated into two parts: one is the green time  $g_{u,d}^s(k_d)$  in which the traffic leaves the link with the saturated flow rate, the other is the green time  $g_{u,d}^d(k_d)$  during which the traffic leaves the link with the arriving flow rate. The quantities of  $g_{u,d}^s(k_d)$  and  $g_{u,d}^d(k_d)$  satisfy the following relationship

$$c_d \alpha_{u,d}^{\text{arriv}}(k_d) + q_{u,d}(k_d) = g_{u,d}^s(k_d) \mu_{u,d} + g_{u,d}^d(k_d) \alpha_{u,d}^{\text{arriv}}(k_d) \quad (7.52)$$

$$g_{u,d}^s(k_d) + g_{u,d}^d(k_d) = g_{u,d}(k_d). \quad (7.53)$$

Hence, we have

$$g_{u,d}^s(k_d) = \frac{c_d \alpha_{u,d}^{\text{arriv}}(k_d) + q_{u,d}(k_d) - g_{u,d}(k_d) \alpha_{u,d}^{\text{arriv}}(k_d)}{\mu_{u,d} - \alpha_{u,d}^{\text{arriv}}(k_d)} \quad (7.54)$$

$$g_{u,d}^d(k_d) = \frac{g_{u,d}(k_d) \mu_{u,d} - c_d \alpha_{u,d}^{\text{arriv}}(k_d) - q_{u,d}(k_d)}{\mu_{u,d} - \alpha_{u,d}^{\text{arriv}}(k_d)}. \quad (7.55)$$

For the unsaturated scenario, the number of vehicles that have behavior  $b \in B$  in link  $(u, d)$  during time period  $[c_d \cdot k_d, c_d \cdot (k_d + 1)]$  is given by

$$N_{u,d}^{\text{free}}(k_d) = n_{u,d}(k_d) - c_d \cdot \alpha_{u,d}^{\text{arriv}}(k_d) - q_{u,d}(k_d) \quad (7.56)$$

$$N_{u,d}^{\text{idling},1}(k_d) = 0 \quad (7.57)$$

$$N_{u,d}^{\text{idling},2}(k_d) = (c_d - g_{u,d}^d(k_d)) \alpha_{u,d}^{\text{arriv}}(k_d) = g_{u,d}^s(k_d) \mu_{u,d} - q_{u,d}(k_d) \quad (7.58)$$

$$N_{u,d}^{\text{idling},3}(k_d) = 0 \quad (7.59)$$

$$N_{u,d}^{\text{idling},4}(k_d) = q_{u,d}(k_d) \quad (7.60)$$

$$N_{u,d}^{\text{dec}}(k_d) = (c_d - g_{u,d}^d(k_d)) \alpha_{u,d}^{\text{arriv}}(k_d) = g_{u,d}^s(k_d) \mu_{u,d} - q_{u,d}(k_d) \quad (7.61)$$

$$N_{u,d}^{\text{acc}}(k_d) = g_{u,d}^s(k_d) \mu_{u,d} \quad (7.62)$$

$$N_{u,d}^{\text{nonstop}}(k_d) = g_{u,d}^d(k_d) \alpha_{u,d}^{\text{arriv}}(k_d), \quad (7.63)$$

and the time periods that the vehicles keep having this behavior in link  $(u, d)$  during time period  $[c_d \cdot k_d, c_d \cdot (k_d + 1)]$  are given by

$$t_{u,d}^{\text{free}}(k_d) = c_d \quad (7.64)$$

$$t_{u,d}^{\text{idling},1}(k_d) = 0 \quad (7.65)$$

$$t_{u,d}^{\text{idling},2}(k_d) = c_d - g_{u,d}^d(k_d) - (v_{\text{low}} - v_{\text{free}})/a_{\text{dec}} \quad (7.66)$$

$$- (v_{\text{free}} - v_{\text{low}})/a_{\text{acc}} \quad (7.67)$$

$$t_{u,d}^{\text{idling},3}(k_d) = 0 \quad (7.68)$$

$$t_{u,d}^{\text{idling},4}(k_d) = c_d - g_{u,d}^d(k_d) - (v_{\text{free}} - v_{\text{low}})/a_{\text{acc}} \quad (7.69)$$

$$t_{u,d}^{\text{dec}}(k_d) = (v_{\text{low}} - v_{\text{free}})/a_{\text{dec}} \quad (7.70)$$

$$t_{u,d}^{\text{acc}}(k_d) = (v_{\text{free}} - v_{\text{low}})/a_{\text{acc}} \quad (7.71)$$

$$t_{u,d}^{\text{nonstop}}(k_d) = g_{u,d}^d(k_d). \quad (7.72)$$

In the unsaturated scenario, no vehicle will be held at the stop-line for more than one cycle time, i.e. all the queues will be dissolved in the following green time. Thus, only vehicles for “idling,2” and “idling,4” exist. All the arriving vehicles except the “nonstop” vehicles (as in (7.58)) will experience deceleration and acceleration, and be idling for the time period in (7.67). All the waiting vehicles in the queues in (7.60) will be idling for the time period (7.69), and then accelerate to leave. Only the arriving vehicles except the vehicles that do not need to stop will decelerate and wait in queues as in (7.61). All the vehicles leaving at the saturation flow rate have to accelerate to leave the link as (7.62) shows. In time period (7.72), the arriving vehicles as shown in (7.63) will leave link  $(u, d)$  without a stop.

## 7.4 Objective function

Given the control time interval  $T_c$  and the simulation time interval  $c_d$  of node  $d \in J$ , there exists an integer  $N_d$  such that

$$T_c = N_d c_d, \quad (7.73)$$

according to the definition of the cycle times for the nodes in a traffic network, as shown in (4.60). For a given  $k_d$  (a counter for simulation time steps for node  $d \in J$ ), the corresponding value of  $k_c$  is given by

$$k_c(k_d) = \left\lfloor \frac{k_d}{N_d} \right\rfloor, \quad (7.74)$$

where  $\lfloor x \rfloor$  for  $x$  a real number denotes the largest integer less than or equal to  $x$ . On the other hand, a given value  $k_c$  of the control time step corresponds to the set  $\{k_c N_d, k_c N_d + 1, \dots, (k_c + 1)N_d - 1\}$  of simulation time steps.

The objective function of the integrated urban control problem at control time step  $k_c$  is

$$J(k_c) = \sum_{\theta \in \Theta} \frac{\lambda_\theta}{E_{\theta, \text{nominal}}} \sum_{(u,d) \in L} \sum_{k_d=N_d k_c+1}^{N_d(k_c+N_p)} E_{\theta, u, d}(k_d), \quad (7.75)$$

where  $E_{\theta, u, d}(k_d)$  denotes the estimated partial criterion for  $\theta$  in link  $(u, d)$  at simulation time step  $k_d$ ,  $\Theta = \{\text{TTS}, \text{CO}, \text{NO}_x, \text{HC}, \text{CO}_2, \text{FC}\}$  is the set of the control objectives,  $E_{\theta, \text{nominal}}$  is the nominal performance for objective  $\theta \in \Theta$  to normalize the partial objective of  $\theta$ , and  $\lambda_\theta$  is the weight parameter for objective  $\theta$ . For the Total Time Spent (TTS), we have

$$E_{\text{TTS}, u, d}(k_d) = T_s \cdot n_{u, d}(k_d), \quad (7.76)$$

and (7.15) will be used for emissions and fuel consumption. The goal of the control problem is to reduce the combined performance of the Total Time Spent and the variety of traffic emissions (i.e. CO, NO<sub>x</sub>, HC, and CO<sub>2</sub>) of the whole urban traffic network over the entire prediction horizon. Hence, it turns out to be a multiple objective control problem. Each of the objective functions is normalized by its nominal performance. By changing the weights of the objective function, a different emphasis can be assigned for different kinds of control purposes.

## 7.5 Case study

CORSIM is a microscopic traffic simulation software developed by FHWA [45], which can be used as a benchmark to design or test traffic control algorithms. We use CORSIM to simulate the real traffic environment, and design MPC controllers according to Chapter 5 to decide control inputs for the traffic signals in CORSIM, as Fig. 7.3 shows. The simulated urban road subnetwork is shown in Fig. 7.4. Nodes marked as ‘‘Sx’’ are the source nodes where traffic flows enter and leave the network. The string S6-6-7-8-9-10-S7 is the main road of the network that has comparatively higher traffic demands. The lightly shaded land in the network is an area of residences or schools, where population density is high, and the requirement for the air condition is also stricter accordingly.

Model predictive controllers for urban traffic are designed to reduce both TTS and TE (Total Emissions for CO, NO<sub>x</sub>, and HC) for this urban traffic network. Given different weights to the partial objective functions, the MPC controller can emphasize on different traffic issues, and focus on improving different performance indications. MPC controllers are designed based on the weights specified for the different objectives, as shown in Table 7.1. Since the CO<sub>2</sub> emission of a vehicle is reduced monotonously when the vehicle speed

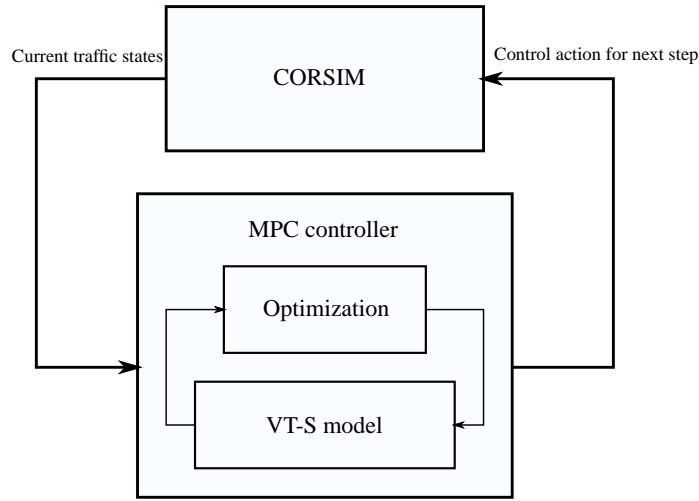


Figure 7.3: Illustration of the traffic control simulation

increases (see Fig. 7.1(d)), we can deduce that the lower time delay is, the higher average traveling speed the vehicles will have, and thus the less CO<sub>2</sub> emission will be released.

The structure of a urban road subnetwork is shown in Fig. 7.4 with a main street (a string S6-6-7-8-9-10-S7) and a resident or a school area (the lightly shaded area). The lengths of the roads are given on the figure in meter. Each of the road in the traffic network have 3 lanes, and the turning rates for each link are all the same, i.e. left turn 33%, through turn 34%, right turn 33%. The storage capacities of the links are fixed according to the link lengths, the number of lanes, and the average vehicle length (7 m). The free-flow speed is 30 km/h. The traffic demands of all the source nodes are very low (500 veh/h), but the traffic demands for source nodes S6 and S7 are very high (3000 veh/h).

For the set-up of the traffic controllers of this subnetwork, the cycle time is set to 60 s for intersection 6, 8, 9, 10, and 11, and 40 s for the other intersections. During the experiments, the simulation time interval of the BLX model is set to 1 s, while in the S model, the simulation time intervals are 60 s or 40 s. For both the MPC controllers, the control time interval  $T_c$  is 120 s, the prediction horizon  $N_p$  is 5, and the control horizon is set to  $N_u = N_p$ . All the simulations implemented with different control strategies run for the same time period, 1 h. To illustrate the effectiveness of the MPC controllers, they are compared with a fixed-time strategy. The fixed-time control strategy is defined having constant phases, cycle times, and green time splits, and the offsets are set to be zero. The fixed-time signals [101] are de-

Table 7.1: Weight parameters for the multi-objective MPC controllers

MPC	$\lambda_{TTS}$	$\lambda_{CO}$	$\lambda_{NO_x}$	$\lambda_{HC}$
MPC-1	1	0	0	0
MPC-2	0	0.33	0.33	0.34
MPC-3	0.5	0.16	0.17	0.17

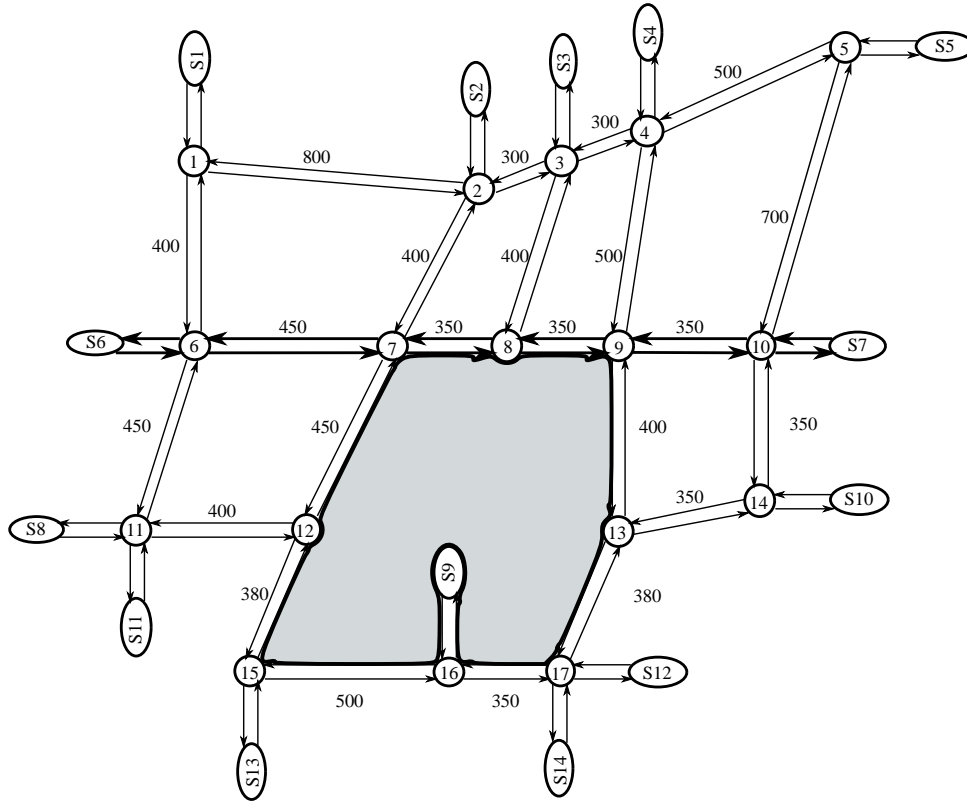


Figure 7.4: An urban road network with a main street (a string S6-6-7-8-9-10-S7) and a resident or a school area (the lightly shaded area)

signed based on the data for the saturated scenario, i.e. the green times are proportional to the traffic demands from each direction, which depend on the saturated flow rates and the turning rates under the saturated scenario.

The performance indicators that CORSIM provides to evaluate the effect of the controllers are the TTS, the Total Fuel Consumption (TFC), the TE for CO, NO<sub>x</sub>, and HC respectively. For the performance TTS, we further compare the TTS for the entire network and the TTS for the main road of the network (i.e. the string S6-6-7-8-9-10-S7, see Fig. 7.4). The TFC represents the total fuel consumption for the whole network, which also reflects the amount of the CO<sub>2</sub> emission (because of the proportional mapping between fuel consumption and CO<sub>2</sub> emission [130], see also (7.9)). The TE is the total amount of gases, including CO, NO<sub>x</sub>, and HC, released on the roads surrounding the lightly shaded region in Fig. 7.4. The fixed-time controller is defined according to the rule described in the answer for Section 5.4.1, which is decided based on the proportion of the saturation flow rates for saturated scenario. But this is not an optimal control solution for real-time traffic, because the traffic demands may change with time, and the traffic scenarios may also switch. The results of every control performance are illustrated in Table 7.2 for different control strategies.



As Table 7.2 shows, all the MPC controllers are able to significantly reduce the TTS, the TFC, and the TE compared to the FT controller, except for the TTS performance of the string. The MPC controllers can improve the TTS performance for the entire traffic network, but also sacrifice some of the TTS performance on the main road. MPC-1 and MPC-2 are MPC controllers taking only the TTS of the whole network or only the TE as control objective respectively. For the MPC-1, the TTS and the TFC are very low, but the TE for each of the gases is higher than that of MPC-2. For the MPC-2, the TE for each of the gases is reduced, but the TTS and the TFC become higher than MPC-1, and the TTS for the string also deteriorates a lot for the main road. When both the TTS and the TE are considered for the control objective as in MPC-3, all the advantages of the MPC-1 and MPC-2 are preserved, where all the performances, including the TTS, the TFC, and the TE of each gas, keep being minimized with only small deviations.

The TTS and TE are not conflicting objectives for urban traffic, but they do conflict to each other for highways, because the speed range and the behavior of vehicles are different for urban and highway. In urban areas, the range of the speed for vehicles is usually from 0 to 50 km/h. We know from the VT-micro model that the emissions and the fuel consumption of a vehicle almost monotonously decrease when the speed of the vehicle increases within this range. Hence, the more congested status the vehicle is in, the more emissions will be released, and the more fuel will be consumed. Therefore, in urban traffic, similar results will be derived when the MPC controller takes TTS and TE as the control objectives. However, this is not the case for highways. The speed of vehicles on highway can go up to 120 veh/h or higher, at which vehicles will also release more emissions and consume more fuel. Therefore, a confliction between TTS and TE will occur in such situation.

## 7.6 Conclusions

In urban areas, traffic delays and traffic emissions are two serious consequences caused by heavy transportation demands. An integrated MPC controller for urban areas is established to reduce both travel delays and various types of traffic emissions.

A new integrated traffic flow and emission model to be used as the prediction model of the MPC controller is proposed in this chapter. The VT-micro emission model for individual vehicles is selected, and integrated with a macroscopic urban traffic flow model, the S-model, so as to form an integrated macroscopic urban traffic flow and emission model, which we call VT-S model. The microscopic emission model can estimate the amount of gases released by a vehicle at different operational condition, i.e. the speed and the acceleration speed. Thus, by combine the traffic flow model and the VT-micro model together, the integrated traffic flow and emission model is able to predict the traffic flow states, as well as the massive emissions released by all vehicles. Moreover, after the emission model is integrated with a macroscopic model, the prediction model can still keep the computational efficiency of a macroscopic model for control purposes. Taking this model as the prediction model, MPC controller can address problems with multiple objectives with respect to both travel delays and emissions. The aim and the preference of the MPC controller can be changed by assigning different weights for the multiple objectives.

This approach was illustrated by a case study. The simulation results show that the MPC controllers can reduce both total emissions and total time spent, and thus reduce total fuel

consumption and CO<sub>2</sub> emissions accordingly.

*Table 7.2: Performance comparison of TTS and TE for a Fixed-time Controller (FT) and for MPC controllers with various objective functions (see Table 7.1)*

Controller	TTS (veh·h)	TTS for string (veh·h)	TFC (l)	TE (kg)		
				CO	NO <sub>x</sub>	HC
FT	1159.8	67.6	832.3	13.1212	0.6164	0.1528
MPC-1	411.1 (-64.6%)	70.7 (4.6%)	488.3 (-41.3%)	9.8302 (-25.08%)	0.5552 (-9.93%)	0.1332 (-12.84%)
MPC-2	447.0 (-61.5%)	78.1 (15.5%)	507.1 (-39.1%)	9.6774 (-26.25%)	0.5477 (-11.14%)	0.1312 (-14.13%)
MPC-3	413.8 (-64.3%)	71.5 (5.6%)	482.3 (-42.1%)	9.7019 (-26.06%)	0.5471 (-11.25%)	0.1312 (-14.10%)

The data in parenthesis are the changing ratio for each performance indicator compared with that of the FT controller



## Chapter 8

# Conclusions and Recommendations

This thesis mainly focuses on how to coordinate and control a large-scale traffic network effectively and in a network-wide way, and it also addresses the efficiency of the computation problem when the network scale grows large.

In this chapter, general conclusions will be drawn, first for the whole thesis, and then for each chapter separately. Furthermore, recommendations for future research directions will be presented.

### 8.1 Summary of conclusions

Model Predictive Control (MPC) is a promising control methodology that can meet the needs for controlling and coordinating a large-scale traffic network with a number of traffic measures. MPC has several advantages for controlling large-scale traffic networks: MPC can easily coordinate various control measures implemented in traffic networks, it can predict the future traffic states to make a long-term decision, it is robust to disturbances and model uncertainty, it can be implemented modularly, etc. However, a big difficulty to implement MPC in practice is the high on-line computational burden. When using MPC, at each time step, we have to solve an optimization problem within a limited period of time. If the optimization problem of MPC controller is too time-consuming to be solved on-line, due to the large scale of the optimization problem or due to the non-linear, non-convex nature of the optimization problem, the MPC controller becomes real-time infeasible in practice, even though the problem is solvable in theory.

Therefore, in the thesis, we have established MPC controllers for urban traffic networks, and presented several ways to address the computational problems arising when MPC is used to control and coordinate large-scale urban traffic networks. The main methods considered in the thesis can be summarized as follows:

- **Model reduction:** The computational efficiency of urban traffic models was improved through reducing the complexity of the models. Accordingly, the on-line optimization problems of the model-based predictive controllers were solved more

efficiently based on these models. In this context, a spatiotemporally discrete urban traffic model with a variable sampling time interval was proposed for model-based predictive control, which allows to search for a trade-off between modeling accuracy and computational complexity. The on-line computational efficiency of the MPC controllers were improved greatly by adjusting the sampling time interval, and accordingly reducing the computational complexity of the prediction models. The control performance loss, caused by the model reduction, can be limited through balancing between the modeling accuracy and the computational complexity of the models.

- **Reformulation of the optimization problem :** We have reformulated the nonlinear non-convex optimization problem of the urban traffic MPC controllers into a mixed-integer linear programming (MILP) problem, where the former is hard to solve by nature, while the latter can be solved efficiently by existing MILP solvers.
- **Hierarchical control structure:** The computational complexity of a centralized MPC controller for a large-scale urban traffic network can be reduced by dividing the network into several smaller sub-networks, each of which results in a much lower computational burden. These traffic sub-networks are coordinated, so as to approximate the global control performance of a centralized MPC controller.

MPC controllers are also built to address multiple control problems for urban traffic networks, e.g. traffic delays as well as vehicle emissions and fuel consumption. In order to control traffic delay as well as traffic emissions and fuel consumption in big cities, an integrated urban traffic, emission, and fuel consumption model is proposed. MPC controllers are established based on this model, which results in a balanced trade-off between minimizing travel time and reducing emissions and fuel consumption.

### Main conclusions for the chapters

- **Chapter 2:**

A literature survey is made to summarize coordinated traffic control strategies for both traffic networks and strings. From the view of traffic control methodologies, the existing coordinated traffic control strategies can be classified into MFD-based (Macroscopic Fundamental Diagram based) approaches, case-based approaches, rule-based approaches, anticipatory control approach, optimal control approaches, and MPC (Model Predictive Control) approaches under centralized, distributed, and hierarchical control structures. The characteristics of these methodologies have been analyzed and compared in this literature survey. The model-based optimization control methodology, including anticipatory control approaches, optimal control approaches, and MPC approaches, is comparatively a very powerful strategy. However, the problem for model-based optimization control methods is the high on-line computational complexity, which handicaps them to be applied in real-life traffic.

- **Chapter 3:**

A well-defined network-wide control framework is necessary for designing controllers for a complex large-scale urban traffic network. By designing a proper control framework for MPC controllers, the on-line computational complexity can be further re-

duced. Moreover, a suitable control structure can also add flexibility, reliability, scalability, sustainability for the complex large-scale traffic control system. Therefore, various control structures for large-scale urban traffic networks were discussed and compared in this chapter. The control structure for a large-scale traffic network can be roughly classified into four types: centralized control structure, decentralized control structure, distributed control structure, and hierarchical control structure. The centralized control can in theory achieve the best overall control performance of the whole network, while the decentralized control has the lowest computational burden. Distributed control and hierarchical control provide a compromise between the centralized controller and the decentralized controller. Therefore, distributed or hierarchical control structures are suitable for controlling complex large-scale urban traffic networks. As a result, a distributed MPC-based control structure was presented, and coordination algorithms were given in this chapter, which provides a general coordination framework for the subnetwork MPC controllers designed in Chapter 5, 6, and 7.

- **Chapter 4:**

Traffic models that can predict future traffic states are the basis of model-based control strategies. The traffic models included in this thesis are all discussed and evaluated from a control point of view. To improve the applicability of the MPC controllers for large-scale urban traffic networks, we are mainly focusing on the trade-off between the efficiency and the complexity of the traffic models. A suitable traffic model that is both accurate enough and fast enough, is very important for model-based controllers from a practical point of view. In this context, several macroscopic urban traffic models were presented in this chapter. Two urban traffic network models, called BLX model and S model respectively, were proposed. The BLX model is more accurate, while the S model is much faster. Moreover, a spatiotemporally discrete urban traffic model with a variable sampling time interval was proposed for model-based predictive control, which allows to easily search for a trade-off between modeling accuracy and computational complexity. It contains the features of both the BLX and the S model. The models analyzed and evaluated in this chapter were used as prediction models of the MPC controllers designed in the later chapters.

- **Chapter 5:**

In this chapter, MPC controllers were established for urban traffic subnetworks, taking respectively the BLX model and the S model proposed in Chapter 4 as prediction model. The S model was selected to further increase the on-line computational speed of the MPC controller. Simulation results illustrated that both the S model and the BLX model were suitable to be used as prediction models of MPC controllers, and that the S model was much faster while still offering acceptable accuracy in the traffic states predictions. Furthermore, the MPC controllers were demonstrated to have sufficient capability for coordinating the traffic control signals and all intersections within the subnetworks and to achieve a good overall performance

- **Chapter 6:**

The on-line computational complexity of MPC controllers was further reduced through reformulating the optimization problem. Due to the nonlinear non-convex nature of

the optimization problem, the on-line computational complexity of the corresponding MPC controllers may become real-time infeasible in practice. To address this problem, the nonlinear  $S$  model was reformulated into a mixed-integer linear model, which can be expressed by mixed-integer linear equalities and inequalities. The  $S$  model and the reduced  $S^*$  model were both reformulated according to this method, and the original nonlinear non-convex optimization problem was written in the form of a Mixed-Integer Linear Programming (MILP) problem based on the reformulated  $S$  model and  $S^*$  model respectively. Hence, existing efficient MILP solvers can be applied to solve the reformulated MILP MPC optimization problems. The simulation experiments illustrated that the MILP-based approaches can maintain a similar control performances as the multi-start SQP-based control approach. Moreover, the most important advantage of the MILP-based MPC controllers is that the on-line computational speed is dramatically increased compared with the original MPC controller, which greatly increases the applicability of the MPC controllers in practice.

- **Chapter 7:**

In order to meet the needs of maintaining a good living environment for citizens, MPC controllers that can consider traffic delay, traffic emissions and fuel consumption were presented in this chapter. An integrated urban traffic, emission, and fuel consumption model was proposed, which combined a macroscopic urban traffic flow model, the  $S$  model, with a microscopic vehicle emission and fuel consumption model, the VT-micro model. This model can predict the future traffic flow states, as well as the emissions released and the fuel consumed by the vehicles. Taking this model as a prediction model, the MPC controller can address problems with multiple objectives with respect to travel delays, emissions, and fuel consumption. Based on this integrated model, we established urban traffic MPC controllers that provide a balanced trade-off between minimizing travel time and reducing both emissions and consumed fuel.

## 8.2 Recommendations for future research

Based on the content of this thesis, we will further give some recommendations on the possible future research directions for both the thesis and the urban traffic control field.

### 8.2.1 Recommendations for the thesis

Due to the advantages of Model Predictive Control (MPC) methods, in this thesis, we applied MPC to control and coordinate urban traffic networks, and tried several approaches to reduce the high on-line computational complexity of MPC controllers. However, although the MPC controllers for urban traffic networks proposed in this thesis are much more efficient than before, it is still not sufficient enough if the scale of a urban traffic becomes too large. Therefore, a well-defined control structure and corresponding proper coordination algorithms are necessary for controlling such a complex large-scale urban traffic network. A general distributed and hierarchical control structure has been presented in the thesis, but the corresponding evaluation and analysis are still missing. Therefore, it is highly recommended to further investigate effective hierarchical and distributed control structures and



algorithms for large-scale urban traffic networks, and to compare the new methods with the presented algorithms. The possible research directions are as follows:

- A **hierarchical control structure** can be built to address the traffic control problems of large-scale traffic networks at different levels. The controller can be specified into multiple layers, on each of which a specific traffic control problem will be solved. The control layers can be defined based on traffic system dynamics (slow changing and fast changing dynamics), characteristics of traffic control problems (traffic control, traffic guidance, etc.), and so on. For each layer, the controller can be established based on a suitable traffic model, which has the best trade-off between descriptive ability and computational complexity, so as to be suitable for the particular control problem of this layer. On the local control level (i.e. intersection level), simple and efficient control methods can be also applied, such as rule-based controllers, PID controllers, fuzzy controllers. Information needs to be exchanged among the layers to achieve a good overall control performance.
- A **distributed control structure** can be also built for controlling and coordinating large-scale traffic networks. A large traffic network can be divided into multiple agents, which can be either traffic sub-networks or even local control actuators (e.g. traffic signals, ramp metering, speed limits, etc.). Agents can exchange information with each other, and can make their own control decisions by their own taking into consideration of the information from other agents. The agents communicate with each other, and work together for a good overall control performance of the whole traffic network. To improve the intelligence of the agents, game theory can be applied to design the communication and coordination algorithms for the agents. In this context, the agents do not only exchange information, but also negotiate with each other, and finally converge to a global equilibrium.

Both the hierarchical control structure and the distributed control structure provide a compromise between the centralized control structure and the decentralized control structure. However, the main challenge for designing hierarchical traffic control structures and algorithms is how to define the control problems of different layers, while the main challenge for designing distributed traffic control structures and algorithms lies on how to exchange the information among subsystems and how to use this information. In addition, before designing the structure and algorithms for hierarchical and distributed controllers, the partitioning of large-scale urban traffic networks into smaller subnetworks need to be investigated.

In Chapter 5, the BLX model and the S model were selected as the prediction models of MPC controllers for urban traffic networks. Actually, there are other urban traffic models that have similar or even better modeling accuracy as the models in this thesis, such as the dynamic network loading model of Bliemer [14], Cell Transmission Model [33] and Link Transmission Model [128] etc. In future, more research on how these models perform when used as prediction models of MPC controllers for urban traffic networks will be carried on.

In Chapter 6, the time delay of the vehicles traveling from the beginning of the link to the end of the queues in the link is constant over time and link. But, in future, this time delay can be further estimated by pre-calibrating queue lengths for different traffic scenarios and environments according to the historical data. In addition, simulations with shorter link

length of an urban traffic network can be carried out to further investigate the dependence between traffic signals and the effect of offsets.

In Chapter 7, an integrated MPC controller was proposed aiming at controlling both travel delays and traffic emissions for urban traffic networks. Vehicle emissions in a particular area, e.g. streets in the neighborhood of schools or residences, were estimated and controlled. However, the dispersion of vehicle emissions was not considered in the thesis. In reality, vehicle emissions will float and spread, after they are released into open air. The spreading direction and speed depends on the current weather, especially the wind. In order to estimate the vehicle emissions in a particular area more accurately, an emission dispersion model, which describes the dispersion dynamics of the emissions after they are released to open air, needs to be further integrated into the traffic flow and traffic emission model. Therefore, the MPC controller based on this model will be able to better limit the vehicle emissions on a school or a residence area more accurately. In addition, the realtime feasibility of the MPC controller in this chapter can become low if the scale of traffic networks increases, thus the optimization problem will be further reformulated into an MILP problem to be solved more efficiently by MILP solvers.

## 8.2.2 Recommendations for the field

In addition, some possible successive research directions can be further considered for urban traffic control field. They can be summarized as follows:

- **Time-varying model-based traffic control.** In fact, an urban traffic network normally is a large-scale system with extreme high complexity. For such a complex traffic network, the model will also become very complicated. In general, both the parameters and the traffic loads of the traffic network model are not static, but will vary with time. However, due to the high complexity, they were rarely considered or were not considered sufficiently, when designing the urban traffic controllers, both in this thesis and in the field. Some of the parameters of the models proposed in the thesis, e.g. turning rates, were assumed to be constant. Actually, these parameters will change with time for different O-D allocations, different route guidance information, different effects of weather conditions, seasonal variations, events (like concerts or soccer games), etc. For instance, the traffic flow turning rates of each intersection will change, if the O-D matrices and traffic route guidance information of the urban traffic network vary from time to time. Moreover, if the variation of the traffic loads for the traffic network model is not considered, the control performance of the MPC controller will also be deteriorated to some extent. Therefore, predictions on the dynamics of the future network traffic loads are also necessary, which can be derived based on the future traffic demand estimation, future dynamic traffic route guidance information, future dynamic road pricing policy, etc. Therefore, time-varying urban traffic models with dynamic network traffic loads predictions can be investigated to adapt both the parameters and the future traffic loads taking into consideration the dynamic traffic information of the network, and MPC controllers can be further built based on these adaptive urban traffic models. In addition, the geographic structure of links, e.g. turning pockets and sources of traffic flows in the middle of a link, can be further considered in urban traffic models.

- **Stochastic-information-based traffic control.** From a microscopic point of view, due to the participation of individual drivers, the traffic states are not only time-varying, but also stochastic. For individual drivers, the travel plan, the time of departure, the route choice, the origin and the destination, are all stochastic, and hard to estimate. But the decisions and choices made by all the individual drivers collectively define the macroscopic traffic phenomena. Therefore, there exists a connection between the microscopic vehicle dynamics and the macroscopic traffic flow dynamics. Traffic control decisions (like traffic signals, ramp metering, variable speed limits, etc.) are made according to the macroscopic traffic flow states. So in this regard, the traffic control decisions also indirectly depend on the dynamics of all the individual vehicles. Therefore, further traffic control strategies can be investigated based on the stochastic traffic information of individual vehicles, which will also take the psychological influences of road users into consideration. In this context, we could also make use of the growing availability of in-car integrated route planning and communication systems to collect more detailed information from individual vehicles and incorporate it to get more accurate predictions of the future traffic evolution.
- **Design of higher-level controllers.** In a hierarchical control structure for a large-scale urban traffic network, controllers can be designed for multiple layers, where at different levels different types of control problems have to be solved. In general, for the higher level control problem, the description of the traffic dynamics can be at a slower rate, the area of the controlled traffic network can be larger, and the traffic model at this level can be more general and less accurate. Normally, high-level controllers will solve an overall control problem with an overview of the whole information of the traffic network, and thus determine set-points, reference trajectories and/or constraints for the low-level controllers. However, these control advices will be only guidelines, which need to be further implemented by the low-level local controllers. In the literature a lot of research is present on low-level controllers for urban traffic networks, but the results for high-level controllers are still partially lacking, in particular regarding performance guarantees, the trade-off between optimality and efficiency, scalability, and robustness. In addition, we need to establish more general traffic models for high-level controllers. For instance, the macroscopic fundamental diagram (MFD) is a very important and useful finding for urban traffic networks, which can be used as a general traffic control model for the high-level urban traffic controllers.
- **High-level models for analysis and design of urban traffic networks.** In the urban traffic control field, control algorithms are mainly evaluated quantitatively by a control performance index, e.g. Total Time Spent (TTS). Normally, the lower the performance index is, the better the control algorithm will function. However, there is lack of qualitative analysis for the cause and effect of the emergence of congestion in large-scale urban traffic networks, especially for the connection between the urban traffic network topology and the emergence of congestion. Of course, this can be also analyzed by means of microscopic simulations, but general conclusions are not easy to derive for different urban traffic networks. Further analysis can be made for investigating more fundamentally the causes and effects of traffic congestion in urban traffic networks. For instance, what kinds of urban traffic network topology will more

easily or more frequently give rise to traffic congestion, which links or intersections of an urban traffic network are the places at which traffic congestion will probably happen first, how will the traffic congestion propagate in an urban traffic network with a specific topology, and so on. More detailed analysis on these topics needs to be performed, and new higher-level or aggregate analysis methods have to be developed. The resulting high-level models can then be used for analysis and design of urban traffic networks and will allow to answer questions regarding which currently uncontrolled intersections to equip with traffic signals, which topology changes to the current network layout have the largest effect on the reduction of congestion, etc.

## Appendix A

# Details for the MILP-based MPC Controller

After the model reformulation, the optimization problem of the MPC controller can be expressed as an MILP problem in the form of (6.40). The optimization problem in (6.40) can be written in detail as follows:

For any link  $(u, d) \in L$  in the network,  $I_{u,d}$  and  $O_{u,d}$  are the set of input nodes and the set of output nodes to link  $(u, d)$  respectively. The inequality constraints in (6.40) are the mixed-integer inequality constraints obtained through Section 6.3.1 made up by the inequality constraints (6.28) and (6.29) for all the traffic streams in the network and for all the predicted simulation time steps in the future (i.e. for  $k_d, k_d + 1, \dots, k_d + MN_p - 1$ ). Here, the inequality constraints for the traffic stream leaving link  $(u, d)$  turning to node  $o \in O_{u,d}$  at time step  $k_d$  are

$$\begin{aligned}
 f_{u,d,o}^1(k_d) &\leq M_{u,d,o}^1 o (1 - \delta_{u,d,o}^1)(k_d) \\
 f_{u,d,o}^1(k_d) &\geq \varepsilon + (m_{u,d,o}^1 - \varepsilon) \delta_{u,d,o}^1(k_d) \\
 z_{u,d,o}^1(k_d) &\leq M_{u,d,o}^1 \delta_{u,d,o}^1(k_d) \\
 z_{u,d,o}^1(k_d) &\geq m_{u,d,o}^1 \delta_{u,d,o}^1(k_d) \\
 z_{u,d,o}^1(k_d) &\leq f_{u,d,o}^1(k_d) - m_{u,d,o}^1 (1 - \delta_{u,d,o}^1(k_d)) \\
 z_{u,d,o}^1(k_d) &\geq f_{u,d,o}^1(k_d) - M_{u,d,o}^1 (1 - \delta_{u,d,o}^1(k_d)) \\
 f_{u,d,o}^2(k_d) &\leq M_{u,d,o}^2 (1 - \delta_{u,d,o}^2(k_d)) \\
 f_{u,d,o}^2(k_d) &\geq \varepsilon + (m_{u,d,o}^2 - \varepsilon) \delta_{u,d,o}^2(k_d) \\
 z_{u,d,o}^2(k_d) &\leq M_{u,d,o}^2 \delta_{u,d,o}^2(k_d) \\
 z_{u,d,o}^2(k_d) &\geq m_{u,d,o}^2 \delta_{u,d,o}^2(k_d) \\
 z_{u,d,o}^2(k_d) &\leq f_{u,d,o}^2(k_d) - m_{u,d,o}^2 (1 - \delta_{u,d,o}^2(k_d)) \\
 z_{u,d,o}^2(k_d) &\geq f_{u,d,o}^2(k_d) - M_{u,d,o}^2 (1 - \delta_{u,d,o}^2(k_d)) .
 \end{aligned} \tag{A.1}$$

In (6.40), the equality constraints for link  $(u, d)$  at time step  $k_d$  are: the linear equations, (4.51) and (4.54), for updating traffic states (the number of vehicles in the link and vehicles

waiting in the queue respectively),

$$n_{u,d}(k_d + 1) = n_{u,d}(k_d) + \alpha_{u,d}^{\text{enter}}(k_d)c_d - \sum_{o \in O_{u,d}} \alpha_{u,d,o}^{\text{leave}}(k_d) \cdot c_d, \quad (\text{A.2})$$

$$q_{u,d,o}(k_d + 1) = q_{u,d,o}(k_d) + \beta_{u,d,o}(k_d)\alpha_{u,d}^{\text{arriv}}(k_d)c_d - \alpha_{u,d,o}^{\text{leave}}(k_d) \cdot c_d, \quad (\text{A.3})$$

the reformulated equations (6.19), (6.22) and (6.27) to substitute the min-max equation of (4.53),

$$f_{u,d,o}^1(k_d) = q_{u,d,o}(k_d)/c_d + \alpha_{u,d,o}^{\text{arriv}}(k_d) - \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_c)/c_d, \quad (\text{A.4})$$

$$f_{u,d,o}^2(k_d) = \beta_{u,d,o}(k_d)(C_{d,o} - n_{d,o}(k_d))/c_d - \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_c)/c_d - z_{u,d,o}^1(k_d) \quad (\text{A.5})$$

$$\alpha_{u,d,o}^{\text{leave}}(k_d) = \beta_{u,d,o}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o}(k_c)/c_d + z_{u,d,o}^1(k_d) + z_{u,d,o}^2(k_d), \quad (\text{A.6})$$

the linearized equation (6.39) to derive the average arriving flow rate to the end of the queues in the link based on Assumption 6.1,

$$\alpha_{u,d}^{\text{arriv}}(k_d) = (1 - \gamma_{\text{const}}) \cdot \alpha_{u,d}^{\text{enter}}(k_d - \tau_{\text{const}}) + \gamma_{\text{const}} \cdot \alpha_{u,d}^{\text{enter}}(k_d - \tau_{\text{const}} - 1), \quad (\text{A.7})$$

the reformulated synchronization equations, (6.33) and (6.38), for the average input flow rate to the link and the number of vehicles in the downstream link of link  $(u, d)$  respectively,

$$\alpha_{i,u,d}^{\text{enter}}(k_d) = \frac{1}{c_d} \sum_{j=0}^{\ell} \xi_{k_d}(k_u + j) \alpha_{i,u,d}^{\text{leave}}(k_u + j), \quad (\text{A.8})$$

$$n_{d,o}(k_d) = \mathcal{F}_{\text{out}}(n_{d,o}^{\text{origin}}(k_o), \dots, n_{d,o}^{\text{origin}}(k_o + \ell)), \quad (\text{A.9})$$

and the cycle time equality constraint for every intersections in the urban traffic network

$$\sum_{p \in P_d} g_p(k_c) = c_d. \quad (\text{A.10})$$

(A.10) is the cycle time constraint which guarantees that the sum of the green time splits of all the phases  $(p \in P_d)$  in intersection  $d$  equals to the cycle time of intersection  $d$ . Except (A.10), all the other equality constraints exist for all the predicted simulation time steps in the future (i.e.  $k_d, k_d + 1, \dots, k_d + MN_p - 1$ ). The cycle time equality constraint (A.10) works only for the future control time steps (i.e.  $k_c, k_c + 1, \dots, k_c + N_p - 1$ ).

# Bibliography

- [1] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59, March 1994.
- [2] K.J. Åström and B. Wittenmark. *Computer-Controlled Systems: Theory and Design*. Prentice Hall, New York, USA, 1996.
- [3] K. Aboudolas, M. Papageorgiou, and E. Kosmatopoulos. Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. *Transportation Research Part C: Emerging Technologies*, 17(2):163–174, 2009.
- [4] K. Ahn, A.A. Trani, H. Rakha, and M. Van Aerde. Microscopic fuel consumption and emission models. In *Proceedings of the 78th Annual Meeting of the Transportation Research Board*, Washington DC, USA, January 1999. CD-ROM.
- [5] K. Ahn, H. Rakha, A. Trani, and M. Van Aerde. Estimating vehicle fuel consumption and emissions based on instantaneous speed and acceleration levels. *Journal of Transportation Engineering*, 128(2):182–190, March/April 2002.
- [6] A. Atamtürk and M.W.P. Savelsbergh. Integer-programming software systems. *Annals of Operations Research*, 140(1):67–124, November 2005.
- [7] C. Audet and J. E. Dennis Jr. Analysis of generalized pattern searches. *SIAM Journal on Optimization*, 13(3):889–903, 2007.
- [8] A. Barisone, D. Giglio, R. Minciardi, and R. Poggi. A macroscopic traffic model for real-time optimization of signalized urban areas. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 900–903, Las Vegas (NV), USA, 2002.
- [9] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.
- [10] D.P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, London, UK, 1982.
- [11] E.M. Bezembinder and F. Brandt. Application of an integrated static and dynamic traffic modelling system for large scale detailed networks. In *Proceedings of the European Transport Conference*, Straatsburg, France, 2004.

- [12] C. Bielefeldt and F. Busch. MOTION—a new on-line traffic signal network control system. In *Proceedings of the 7th International Conference on Road Traffic Monitoring and Control*, pages 55–59, London, UK, April 1994.
- [13] C. Bielefeldt and F. Busch. MOTION—a new on-line traffic signal network control system. In *Proceedings of the 7th International Conference on Road Traffic Monitoring and Control*, pages 55–59. IET, 2002.
- [14] M.C.J. Bliemer. Dynamic Queuing and Spillback in Analytical Multiclass Dynamic Network Loading Model. *Transportation Research Record: Journal of the Transportation Research Board*, 2029(-1):14–21, 2007. ISSN 0361-1981.
- [15] K. Bogenberger, H. Keller, and A. D. May. A neuro-fuzzy approach for ramp metering. In *Proceedings of the 10th International Conference on Road Transport Information and Control*, pages 101–105, London, UK, 2000.
- [16] K. Bogenberger, S. Vukanovic, and H. Keller. ACCEZZ — adaptive fuzzy algorithms for traffic responsive and coordinated ramp metering. In *Proceedings of the Applications of Advanced Technology in Transportation 2002*, pages 94–94, Boston (MA), USA, 2002.
- [17] F. Boillot, J.M. Blossville, J.B. Lesort, V. Motyka, M. Papageorgiou, and S. Sellam. Optimal signal control of urban traffic networks. In *Proceedings of the Conference on Road Traffic Monitoring and Control*, pages 75–79, 1992.
- [18] F. Boillot, S. Midenet, and J. C. Pierrelee. The real-time urban traffic control system CRONOS: Algorithm and experiments. *Transportation Research Part C: Emerging Technologies*, 14(1):18–38, 2006.
- [19] P.G. Boulter, T. Barlow, I.S. McCrae, S. Latham, D. Elst, and E. Van der Burgwal. Road traffic characteristics, driving patterns and emission factors for congested situations. Technical report, TNO Automotive, Department Powertrains-Environmental Studies and Testing, Delft, The Netherlands, 2002.
- [20] S.P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.
- [21] D. Bretherton, M. Bodger, and N. Baber. SCOOT—the future urban traffic control. In *Proceedings of the 12th IEEE International Conference on Road Transport Information and Control*, pages 301–306, London, UK, April 2004.
- [22] G.W. Brown. Iterative solution of games by fictitious play. *Activity Analysis of Production and Allocation*, 13(1):374–376, 1951.
- [23] E.F. Camacho and C. Bordons. *Model Predictive Control in the Process Industry*. Springer-Verlag, Berlin, Germany, 1995.
- [24] E. Camponogara, D. Jia, B.H. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, 22(1):44–52, 2002.



- [25] S.F. Cheng, M. A. Epelman, and R. L. Smith. CoSIGN: A parallel algorithm for coordinated traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):551–564, 2006.
- [26] M.C. Choy, D. Srinivasan, and R.L. Cheu. Cooperative, hybrid agent architecture for real-time traffic signal control. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 33(5):597–607, 2003.
- [27] D. Christiansen. *Electronics Engineers' Handbook*. IEEE Press/McGraw Hill, New York, USA, 4th edition, 1997.
- [28] M.C. Coelho, T.L. Farias, and N.M. Roupail. Impact of speed control traffic signals on pollutant emissions. *Transportation Research Part D*, 10(4):323–340, July 2005.
- [29] M.C. Coelho, T.L. Farias, and N.M. Roupail. Effect of roundabout operations on pollutant emissions. *Transportation Research Part D*, 11(5):333–343, September 2006.
- [30] R. Corthout, C.M.J. Tampère, Rodric. Frederix, and L.H. Immers. Marginal dynamic network loading for large-scale simulation-based applications. In *Proceedings of the 90th Annual Meeting of the Transportation Research Board*, Washington D.C., USA, 2011. Paper 11-0681.
- [31] R. Courant, K. Friedrichs, and H. Lewy. On the partial difference equations of mathematical physics. *IBM Journal of Research and Development*, 11(2):215–234, 1967.
- [32] C.F. Daganzo. The cell transmission model, part ii: Network traffic. *Transportation Research Part B: Methodological*, 29(2):79 – 93, 1995.
- [33] C.F. Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4):269–287, 1994. ISSN 0191-2615.
- [34] C.F. Daganzo. Requiem for second-order fluid approximations of traffic flow. *Transportation Research Part B*, 29(4):277–286, 1995.
- [35] C.F. Daganzo and N. Geroliminis. An analytical approximation for the macroscopic fundamental diagram of urban traffic. *Transportation Research Part B: Methodological*, 42(9):771–781, 2008.
- [36] L. Davis, editor. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, USA, 1991.
- [37] B. De Schutter and T.J.J. van den Boom. Model predictive control for discrete-event and hybrid systems – Part II: Hybrid systems. In *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2004)*, Leuven, Belgium, July 2004. Paper 313.
- [38] B. De Schutter, S.P. Hoogendoorn, H. Schuurman, and S. Stramigioli. A multi-agent case-based traffic control scenario evaluation system. In *Proceedings of the IEEE 6th International Conference on Intelligent Transportation Systems (ITSC'03)*, pages 678–683, Shanghai, China, October 2003.

- [39] A. Di Febbraro, D. Giglio, and N. Sacco. Urban traffic control structure based on hybrid petri nets. *IEEE Transactions on Intelligent Transportation Systems*, 5(4): 224–237, Dec. 2004.
- [40] C. Diakaki, M. Papageorgiou, and K. Aboudolas. A multivariable regulator approach to traffic-responsive network-wide signal control. *Control Engineering Practice*, 10(2):183–195, 2002.
- [41] V. Dinopoulou, C. Diakaki, and M. Papageorgiou. Applications of the urban traffic control strategy TUC. *European Journal of Operational Research*, 175(3):1652–1665, 2006.
- [42] M. Dotoli, M. P. Fanti, and C. Meloni. A signal timing plan formulation for urban traffic control. *Control Engineering Practice*, 14(11):1297–1311, 2006.
- [43] R.W. Eglese. Simulated annealing: A tool for operations research. *European Journal of Operational Research*, 46(3):271–281, June 1990.
- [44] J.L. Farges, J.J. Henry, and J. Tufal. The PRODYN real-time traffic algorithm. In *Proceedings of the 4th IFAC Symposium of Transportation Systems*, pages 307–312, Baden Baden, Germany, 1983.
- [45] FHWA. *Traffic Software Integrated System Version 5.1 User's Guide*. 2001.
- [46] R. Fletcher and S. Leyffer. Numerical experience with lower bounds for MIQP branch-and-bound. *SIAM Journal on Optimization*, 8(2):604–616, May 1998.
- [47] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, San Francisco (CA), USA, 1979.
- [48] N. H. Gartner, F. J. Pooran, and C. M. Andrews. Implementation of the OPAC adaptive control strategy in a traffic signal network. In *Proceedings of the 2001 IEEE International Intelligent Transportation Systems Conference*, pages 195–200, Oakland (CA), USA, 2001.
- [49] N.H. Gartner. Simulation study of OPAC: A demand-responsive strategy for traffic signal control. *Transportation and Traffic Theory*, pages 233–250, 1983.
- [50] N.H. Gartner and C. Stamatiadis. Integration of dynamic traffic assignment with real-time traffic adaptive control system. *Transportation Research Record: Journal of the Transportation Research Board*, (1644):150–156, 1998.
- [51] N.H. Gartner and C. Stamatiadis. Arterial-based control of traffic flow in urban grid networks. *Mathematical and computer modelling*, 35(5-6):657–672, 2002.
- [52] D.C. Gazis and R.B. Potts. The oversaturated intersection. In *Proceedings of the 2nd International Symposium on Traffic Theory*, 1963.
- [53] N. Geroliminis and C.F. Daganzo. Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. *Transportation Research Part B: Methodological*, 42(9):759–770, 2008.

- [54] A.H. Ghods and A. Rahimi-Kian. A game theory approach to optimal coordinated ramp metering and variable speed limits. In *Proceedings of the 2008 Chinese Control and Decision Conference*, pages 91–96, Yantai, China, 2008.
- [55] G. Gomes, R. Horowitz, A.A. Kurzhanskiy, P. Varaiya, and J. Kwon. Behavior of the cell transmission model and effectiveness of ramp metering. *Transportation Research Part C: Emerging Technologies*, 16(4):485–513, 2008.
- [56] D.K. Hale. Traffic Network Study Tool, TRANSYT-7F, United States Version. *McTrans Center in the University of Florida*, 2005.
- [57] F. Hayes-Roth. Rule-based systems. *Communications of the ACM*, 28(9):921–932, 1985.
- [58] A. Hegyi. *Model Predictive Control for Integrating Traffic Control Measures*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2004.
- [59] A. Hegyi, B. De Schutter, and J. Hellendoorn. Optimal coordination of variable speed limits to suppress shock waves. *IEEE Transactions on Intelligent Transportation Systems*, 6(1):102–112, 2005.
- [60] A. Hegyi, B. De Schutter, and J. Hellendoorn. Model predictive control for optimal coordination of ramp metering and variable speed limits. *Transportation Research Part C: Emerging Technologies*, 13(3):185–209, 2005.
- [61] T.H. Heung, T.K. Ho, and Y.F. Fung. Coordinated road-junction traffic control by dynamic programming. *IEEE Transactions on Intelligent Transportation Systems*, 6(3):341–350, 2005.
- [62] J. Heywood. *Internal Combustion Engine Fundamentals*. McGrawHill, New York, 1988.
- [63] S.P. Hoogendoorn and P.H.L. Bovy. State-of-the-art of vehicular traffic flow modelling. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 215(4):283–303, 2001. ISSN 0959-6518.
- [64] S.P. Hoogendoorn, B. De Schutter, and H. Schuurman. Decision support in dynamic traffic management. Real-time scenario evaluation. *European Journal of Transport and Infrastructure Research*, 3(1):21–38, 2003.
- [65] S.P. Hoogendoorn, H. Schuurman, and B. De Schutter. Real-time traffic management scenario evaluation. In *Proceedings of the 10th IFAC Symposium on Control in Transportation Systems (CTS 2003)*, pages 343–348, Tokyo, Japan, August 2003.
- [66] H.R. Kashani and G.N. Saridis. Intelligent control for urban traffic systems. *Automatica*, 19(2):191–197, 1983.
- [67] G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, New York, USA, 1995.

- [68] E. Kosmatopoulos, M. Papageorgiou, C. Bielefeldt, V. Dinopoulou, R. Morris, J. Mueck, A. Richards, and F. Weichenmeier. International comparative field evaluation of a traffic-responsive signal control strategy in three cities. *Transportation Research Part A: Policy and Practice*, 40(5):399–413, 2006.
- [69] A. Kotsialos and M. Papageorgiou. Motorway network traffic control systems. *European Journal of Operational Research*, 152(2):321–333, 2004.
- [70] A. Kotsialos and M. Papageorgiou. Efficiency and equity properties of freeway network-wide ramp metering with AMOC. *Transportation Research Part C: Emerging Technologies*, 12(6):401–420, 2004.
- [71] A. Kotsialos and M. Papageorgiou. Nonlinear optimal control applied to coordinated ramp metering. *IEEE Transactions on Control Systems Technology*, 12(6):920–933, 2004.
- [72] A. Kotsialos, M. Papageorgiou, M. Mangeas, and H. Haj-Salem. Coordinated and integrated control of motorway networks via non-linear optimal control. *Transportation Research Part C: Emerging Technologies*, 10(1):65–84, 2002.
- [73] A. Kotsialos, M. Papageorgiou, and F. Middelham. Local and optimal coordinated ramp metering for freeway networks. *Journal of Intelligent Transportation Systems*, 9(4):187–203, 2005.
- [74] B. Krause and C. von Altrock. 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*, Berlin, Germany, October 1997. Paper no. 2045.
- [75] R.H. Krikke. HARS: The next generation traffic control system. In *Proceedings of the 11th IFAC Symposium on Control in Transportation Systems*, Delft, The Netherlands, August 2006.
- [76] T.J. Lambert III, M.A. Epelman, and R.L. Smith. A fictitious play approach to large-scale optimization. *Operations Research-Linthicum*, 53(3):477–489, 2005.
- [77] J.P. Lebacque. The Godunov scheme and what it means for first order traffic flow models. In *Proceedings of the International Symposium on Transportation and Traffic Theory*, pages 647–677, Lyon, France, 1996.
- [78] D.W. Li and Y.G. Xi. The general framework of aggregation strategy in model predictive control and stability analysis. In *Proceedings of the 11th IFAC Symposium on Large Scale Systems Theory and Applications*, Gdansk, Poland, 2007.
- [79] M.J. Lighthill and G.B. Whitham. On kinematic waves. II. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 229(1178):317–345, 1955.
- [80] S. Lin and Y. Xi. An efficient model for urban traffic network control. In *Proceedings of the 17th World Congress The International Federation of Automatic Control*, pages 14066–14071, Seoul, Korea, July 2008.

- [81] S. Lin, Y. Xi, and Y. Yang. Short-term traffic flow forecasting using macroscopic urban traffic network model. In *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems*, pages 134–138, Beijing, China, October 2008.
- [82] S. Lin, B. De Schutter, Y. Xi, and J. Hellendoorn. A simplified macroscopic urban traffic network model for model-based predictive control. In *Proceedings of the 12th IFAC Symposium Control Transportation Systems*, pages 286–291, Redondo Beach (CA), USA, September 2009.
- [83] S. Lin, B. De Schutter, Y. Xi, and J. Hellendoorn. Study on fast model predictive controllers for large urban traffic networks. In *Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems*, pages 691–696, St. Louis (MO), USA, October 2009.
- [84] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn. Model predictive control for urban traffic networks via MILP. In *Proceedings of the 2010 American Control Conference*, pages 2272–2277, Baltimore (MA), USA, June–July 2010.
- [85] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn. An efficient model-based method for coordinated control of urban traffic networks. In *Proceedings of the 2010 IEEE International Conference on Networking, Sensing and Control*, pages 8–13, Chicago (IL), Illinois, April 2010.
- [86] S. Lin, B. De Schutter, S.K. Zegeye, H. Hellendoorn, and Y. Xi. Integrated urban traffic control for the reduction of travel delays and emissions. In *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC 2010)*, pages 677–682, Madeira Island, Portugal, September 2010.
- [87] S. Lin, B. De Schutter, A. Hegyi, Y. Xi, and H. Hellendoorn. Spatiotemporally discrete urban traffic model. In *Accepted for publication in 18th World Congress The International Federation of Automatic Control*, Milano, Italy, July 2011.
- [88] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn. Fast model predictive control for urban road networks via milp. *IEEE Transactions on Intelligent Transportation Systems*, PP(99):1–11, 2011.
- [89] J.T. Linderoth and T.K. Ralphs. Noncommercial software for mixed-integer linear programming. *Integer Programming: Theory and Practice*, pages 253–303, 2005.
- [90] H.K. Lo, E. Chang, and Y.C. Chan. Dynamic network traffic control. *Transportation Research Part A: Policy and Practice*, 35(8):721–744, 2001.
- [91] P.R. Lowrie. The sydney coordinated adaptive traffic system: Principles, methodology, algorithms. In *Proceedings of the International Conference on Road Traffic Signalling*, pages 67–70, 1982.
- [92] J.M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, Harlow, UK, 2002.

- [93] V. Mauro and C. Di Taranto. Utopia. In *Proceedings of the 2nd IFAC-IFIP-IFORS Symposium on Traffic Control and Transportation Systems*, pages 575–597, 1989.
- [94] R.D. McKelvey and A. McLennan. Computation of equilibria in finite games. *Handbook of Computational Economics*, 1:87–142, 1996.
- [95] M.D. Mesarovic, D. Macko, and Y. Takahara. Two coordination principles and their application in large scale systems control. *Automatica*, 6(2):261–270, 1970.
- [96] A. Messmer and M. Papageorgiou. METANET: A macroscopic simulation program for motorway networks. *Traffic Engineering and Control*, 31(8-9):466–70, 1990.
- [97] P. Mirchandani and L. Head. A real-time traffic signal control system: Architecture, algorithm, and analysis. *Transportation Research Part C: Emerging Technologies*, 9(6):415–432, 2001.
- [98] D. Monderer and L.S. Shapley. Fictitious play property for games with identical interests. *Journal of Economic Theory*, 68(1):258–265, 1996.
- [99] R.R. Negenborn, B. De Schutter, and J. Hellendoorn. Multi-agent model predictive control for transportation networks: Serial versus parallel schemes. *Engineering Applications of Artificial Intelligence*, 21(3):353–366, April 2008.
- [100] H. T. Nguyen and E. A. Walker. *A First Course in Fuzzy Logic*. RCS Publications, 2nd edition, 1999.
- [101] M. Papageorgiou. *Applications of Automatic Control Concepts to Traffic Flow Modeling and Control*. Springer-Verlag New York, Secaucus (NJ), USA, 1983.
- [102] M. Papageorgiou. Some remarks on macroscopic traffic flow modelling. *Transportation Research Part A: Policy and Practice*, 32(5):323–329, 1998.
- [103] M. Papageorgiou, H. Hadj-Salem, and J.M. Blosseville. ALINEA: A local feedback control law for on-ramp metering. *Transportation Research Record*, (1320), 1991.
- [104] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12):2043–2067, 2003.
- [105] I. Papamichail and M. Papageorgiou. Traffic-responsive linked ramp-metering control. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):111–121, 2008.
- [106] P.M. Pardalos and M.G.C. Resende, editors. *Handbook of Applied Optimization*. Oxford University Press, Oxford, UK, 2002.
- [107] H.J. Payne. Models of freeway traffic and control. *Mathematical Models of Public Systems*, 1(1):51–61, 1971.
- [108] J.B. Rawlings and D.Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison (WI), USA, 2009.
- [109] P.I. Richards. Shock waves on the highway. *Operations Research*, 4(1):42–51, 1956.

- [110] N. Lawrence Ricker. Use of quadratic programming for constrained internal model control. *Industrial and Engineering Chemistry Process Design and Development*, 24(4):925–936, 1985.
- [111] S. G. Ritchie. A knowledge-based decision support architecture for advanced traffic management. *Transportation Research Part A*, 24(1):27–37, January 1990.
- [112] D.I. Robertson and R.D. Bretherton. Optimizing networks of traffic signals in real time - The SCOOT method. *IEEE Transactions on Vehicular Technology*, 40(1):11–15, 1991.
- [113] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs (NE), USA, 2003.
- [114] R. Scattolini. Architectures for distributed and hierarchical Model Predictive Control- A review. *Journal of Process Control*, 19(5):723–731, 2009.
- [115] S. Sen and K.L. Head. Controlled optimization of phases at an intersection. *Transportation Science*, 31(1):5–17, 1997.
- [116] H. Taale. *Integrated Anticipatory Control of Road Networks - A game-theoretical approach*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2008.
- [117] H. Taale and F. Middelham. Ten years of ramp-metering in The Netherlands. In *Proceedings of the 10th International Conference on Road Transport Information and Control*, pages 106–110, London, UK, April 2000.
- [118] H. Taale and H.J. van Zuylen. Traffic control and route choice: Occurrence of instabilities. In *Proceedings of the 5th TRAIL Congress: Five Years Crossroads of Theory and Practice*, volume 2, pages 1–19, Rotterdam, The Netherlands, 1999.
- [119] M. van den Berg, A. Hegyi, B. De Schutter, and J. Hellendoorn. A macroscopic traffic flow model for integrated control of freeway and urban traffic networks. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 2774–2779, Maui (HI), USA, 2003.
- [120] M. van den Berg, B. De Schutter, A. Hegyi, and J. Hellendoorn. Model predictive control for mixed urban and freeway networks. In *Proceedings of the 83rd Annual Meeting of the Transportation Research Board*, Washington D.C., USA, 2004. Paper 04-3327.
- [121] M. van den Berg, A. Hegyi, B. De Schutter, and J. Hellendoorn. Integrated traffic control for mixed urban and freeway networks: A model predictive control approach. *European Journal of Transport and Infrastructure Research*, 7(3):223–250, 2007.
- [122] R.T. Van Katwijk. *Multi-Agent Look-Ahead Traffic-Adaptive Control*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2008.
- [123] R.T. van Katwijk, P. van Koningsbruggen, B. De Schutter, and J. Hellendoorn. A test bed for multi-agent control systems in road traffic management. In F. Klügl, A. Bazzan, and S. Ossowski, editors, *Applications of Agent Technology in Traffic and*

- Transportation*, Whitestein Series in Software Agent Technologies, pages 113–131. Birkhäuser Verlag, Basel, Switzerland, 2005.
- [124] J. Vrancken and F. Ottenhof. A bottom-up approach to implementing sustainable traffic control. In *Proceedings of the 11th IFAC Symposium on Control in Transportation Systems*, Delft, The Netherlands, August 2006.
- [125] J. Vrancken, F. Ottenhof, and M. Scares. Regional road traffic management in The Netherlands. *Urban Transport XIII: Urban Transport and the Environment in the 21st Century*, 96:275–282, 2007.
- [126] S. Vukanovic and O. Ernhof. Evaluation and field implementation of the fuzzy logic based ramp metering algorithm *acezz*. In *Proceedings of the 9th IEEE Conference on Intelligent Transportation Systems*, pages 437–441, Toronto, Canada, 2006.
- [127] F.V. Webster. Traffic signals. *Traffic Engineering Practice*, pages 117–121, 1963.
- [128] I. Yperman, S. Logghe, C.M.J. Tampère, and L.H. Immers. The Multi-Commodity Link Transmission Model for Dynamic Network Loading. In *Proceedings of the 85th Annual Meeting of the Transportation Research Board*, Washington D.C., 2006.
- [129] S.K. Zegeye, B. De Schutter, H. Hellendoorn, and E. Breunese. Reduction of travel times and traffic emissions using model predictive control. In *Proceedings of the 2009 American Control Conference*, pages 5392–5397, St. Louis (MO), USA, June 2009.
- [130] S.K. Zegeye, B. De Schutter, H. Hellendoorn, and E. Breunese. Model-based traffic control for balanced reduction of fuel consumption, emissions, and travel time. In *Proceedings of the 12th IFAC Symposium on Transportation Systems*, pages 149–154, Redondo Beach (CA), USA, September 2009.
- [131] S.K. Zegeye, B. De Schutter, J. Hellendoorn, and E.A. Breunese. Model-based traffic control for the reduction of fuel consumption, emissions, and travel time. In *Proceedings of the mobil.TUM 2009 — International Scientific Conference on Mobility and Transport*, Munich, Germany, May 2009.



# TRAIL Thesis Series

The following list contains the most recent dissertations in the TRAIL Thesis Series. For a complete overview of more than 100 titles see the TRAIL website: [www.rsTRAIL.nl](http://www.rsTRAIL.nl).

The TRAIL Thesis Series is a series of the Netherlands TRAIL Research School on transport, infrastructure and logistics.

Lin, S., *Efficient Model Predictive Control for Large-Scale Urban Traffic Networks*, T2011/3, April 2011, TRAIL Thesis Series, the Netherlands

Oort, N. van, *Service Reliability and Urban Public Transport Design*, T2011/2, April 2011, TRAIL Thesis Series, the Netherlands

Mahmod, M.K.M., *Using Co-Operative Vehicle-Infrastructure Systems to Reduce Traffic Emissions and Improve Air Quality at Signalized Urban Intersections*, T2011/1, March 2011, TRAIL Thesis Series, the Netherlands

Corman, F., *Real-Time Railway Traffic Management: dispatching in complex, large and busy railway networks*, T2010/14, December 2010, TRAIL Thesis Series, the Netherlands

Kwakkel, J., *The Treatment of Uncertainty in Airport Strategic Planning*, T2010/13, December 2010, TRAIL Thesis Series, the Netherlands

Pang, Y., *Intelligent Belt Conveyor Monitoring and Control*, T2010/12, December 2010, TRAIL Thesis Series, the Netherlands

Kim, N.S., *Intermodal Freight Transport on the Right Track? Environmental and economic performances and their trade-off*, T2010/11, December 2010, TRAIL Thesis Series, the Netherlands

Snelder, M., *Designing Robust Road Networks: a general design method applied to the Netherlands*, T2010/10, December 2010, TRAIL Thesis Series, the Netherlands

Hinsbergen, C.P.IJ. van, *Bayesian Data Assimilation for Improved Modeling of Road Traffic*, T2010/9, November 2010, TRAIL Thesis Series, the Netherlands

Zuurbier, F.S., *Intelligent Route Guidance*, T2010/8, November 2010, TRAIL Thesis Series, the Netherlands

Larco Martinelli, J.A., *Incorporating Worker-Specific Factors in Operations Management Models*, T2010/7, November 2010, TRAIL Thesis Series, the Netherlands

Ham, J.C. van, *Zeehavenontwikkeling in Nederland: naar een beter beleidsvormingsproces*, T2010/6, August 2010, TRAIL Thesis Series, the Netherlands

Boer, E. de, *School Concentration and School Travel*, T2010/5, June 2010, TRAIL Thesis

Series, the Netherlands

Berg, M. van den, *Integrated Control of Mixed Traffic Networks using Model Predictive Control*, T2010/4, April 2010, TRAIL Thesis Series, the Netherlands

Top, J. van den, *Modelling Risk Control Measures in Railways*, T2010/3, April 2010, TRAIL Thesis Series, the Netherlands

Craen, S. de, *The X-factor: a longitudinal study of calibration in young novice drivers*, T2010/2, March 2010, TRAIL Thesis Series, the Netherlands

Tarau, A.N., *Model-based Control for Postal Automation and Baggage Handling*, T2010/1, January 2010, TRAIL Thesis Series, the Netherlands

Knoop, V.L., *Road Incidents and Network Dynamics: effects on driving behaviour and traffic congestion*, T2009/13, December 2009, TRAIL Thesis Series, the Netherlands

Baskar, L.D., *Traffic Control and Management with Intelligent Vehicle Highway Systems*, T2009/12, November 2009, TRAIL Thesis Series, the Netherlands

Konings, J.W., *Intermodal Barge Transport: network design, nodes and competitiveness*, T2009/11, November 2009, TRAIL Thesis Series, the Netherlands

Kusumaningtyas, I., *Mind Your Step: exploring aspects in the application of long accelerating moving walkways*, T2009/10, October 2009, TRAIL Thesis Series, the Netherlands

Gong, Y., *Stochastic Modelling and Analysis of Warehouse Operations*, T2009/9, September 2009, TRAIL Thesis Series, the Netherlands

Eddia, S., *Transport Policy Implementation and Outcomes: the case of Yaounde in the 1990s*, T2009/8, September 2009, TRAIL Thesis Series, the Netherlands

Platz, T.E., *The Efficient Integration of Inland Shipping into Continental Intermodal Transport Chains: measures and decisive factors*, T2009/7, August 2009, TRAIL Thesis Series, the Netherlands

Tahmasseby, S., *Reliability in Urban Public Transport Network Assessment and Design*, T2009/6, June 2009, TRAIL Thesis Series, the Netherlands

Bogers, E.A.I., *Traffic Information and Learning in Day-to-day Route Choice*, T2009/5, June 2009, TRAIL Thesis Series, the Netherlands

Amelsfort, D.H. van, *Behavioural Responses and Network Effects of Time-varying Road Pricing*, T2009/4, May 2009, TRAIL Thesis Series, the Netherlands

Li, H., *Reliability-based Dynamic Network Design with Stochastic Networks*, T2009/3, May 2009, TRAIL Thesis Series, the Netherlands

Stankova, K., *Stackelberg and Inverse Stackelberg Games & their Applications in the Optimal Toll Design Problem, the Energy Markets Liberalization Problem, and in the Theory of Incentives*, T2009/2, February 2009, TRAIL Thesis Series, the Netherlands

# Samenvatting

## Efficiënte modelgebaseerde voorspellende regeling voor grootschalige stedelijke verkeersnetwerken

Congestie in stedelijke omgevingen, in het bijzonder in grote steden, vormt een uitdagend probleem. Een veelbelovende regelstrategie die voldoet aan de meeste vereisten die nodig zijn voor de gecoördineerde regeling van grootschalige stedelijke verkeersnetwerken, is modelgebaseerde voorspellende regeling (*Model Predictive Control*, MPC). MPC is een regelstrategie die gebaseerd is op optimalisatie en die optimale regelingen kan bepalen voor systemen met verscheidene ingangs- en uitgangssignalen met zowel lineaire als niet-lineaire beperkingen. MPC biedt een aantal voordelen voor het regelen van grootschalige verkeersnetwerken: MPC kan op eenvoudige wijze verschillende verkeersmaatregelen coördineren, MPC kan de toekomstige toestand van het verkeersnetwerk voorspellen en zo lange-termijn-beslissingen nemen, MPC is robuust voor verstoringen en modelonzekerheden en MPC kan modulair geïmplementeerd worden. Daarom is MPC voor stedelijke verkeersnetwerken als onderwerp van dit proefschrift gekozen.

Het belangrijkste probleem om MPC in de praktijk te implementeren is de hoge online rekentijd die vereist is. Bij MPC moet immers voor elke tijdstap in korte tijd een optimalisatieprobleem opgelost worden. Als ten gevolge van de grootschaligheid van het optimalisatieprobleem of van het niet-lineaire, niet-convexe karakter van dit probleem het oplossen van het MPC-optimalisatieprobleem te veel tijd vergt om online opgelost te kunnen worden, wordt de MPC-regelaar in de praktijk onuitvoerbaar in real-time, hoewel het optimalisatieprobleem theoretisch gezien wel een oplossing kan hebben. De online rekenefficiëntie van MPC-regelaars is het kernprobleem dat opgelost moet worden, voordat MPC toegepast kan worden in grootschalige stedelijke verkeersnetwerken.

In dit proefschrift stellen we verscheidene manieren voor om de rekenproblemen aan te pakken die ontstaan wanneer MPC wordt gebruikt voor het regelen en coördineren van grootschalige stedelijke verkeersnetwerken. De belangrijkste methoden die beschouwd worden in dit proefschrift, kunnen als volgt worden samengevat:

- **Modelreductie:** Vanwege het niet-lineaire karakter van modellen voor stedelijk verkeer wordt het MPC-optimalisatieprobleem een niet-lineair, niet-convex optimalisatieprobleem. Zo'n probleem kan opgelost worden door niet-lineaire optimalisatie-algoritmen, die echter een groot aantal evaluaties van de doelfunctie door middel van het voorspellingsmodel vereisen. Dit betekent dat hoe complexer het gebruikte model voor stedelijk verkeer is, hoe meer rekentijd nodig is om het online optimalisatiepro-

ces uit te voeren. Bijgevolg kan de online rekenefficiëntie van MPC verbeterd worden door de complexiteit van het voorspellingsmodel dat in de MPC-regelaar gebruikt wordt te reduceren.

- **Herformulering van het optimalisatieprobleem:** Vanwege het niet-lineaire, niet-convexe karakter van het MPC-optimalisatieprobleem zal de rekencomplexiteit in de praktijk min of meer exponentieel toenemen als de omvang van het geregelde verkeersnetwerk toeneemt. Alhoewel de online rekenefficiëntie van de MPC-regelaar verbeterd kan worden door de complexiteit van het gebruikte voorspellingsmodel te reduceren, is het effect daarvan nog steeds beperkt voor grote verkeersnetwerken. Om dit te vermijden kan de online rekenefficiëntie ook verbeterd worden door het niet-lineaire, niet-convexe MPC-optimalisatieprobleem te herformuleren als een optimalisatieprobleem dat veel efficiënter opgelost kan worden.
- **Hiërarchische regeling:** De rekencomplexiteit van een gecentraliseerde MPC-regelaar voor een grootschalig stedelijk verkeersnetwerk kan verminderd worden door het netwerk op te delen in verscheidene deelnetwerken, die dan resulteren in een veel lagere rekenbelasting. Deze deelnetwerken moeten dan wel gecoördineerd worden om de globale regelprestatie van een gecentraliseerde MPC-regelaar zo goed mogelijk te benaderen.

In dit proefschrift worden in het bijzonder de eerste twee methodes onderzocht, namelijk „modelreductie” en „herformulering van het optimalisatieprobleem”. Wij tonen aan dat de online rekentijd voor de MPC-optimalisatieproblemen door deze twee methodes gereduceerd wordt. Voor de derde methode presenteren we een algemeen framework voor de MPC-regeling van grootschalige stedelijke verkeersnetwerken en analyseren we de verschillende regelstructuren voor netwerkbrede verkeersregeling. Alle gecentraliseerde MPC-regelaars die in dit proefschrift worden voorgesteld, passen in het framework en kunnen gebruikt worden als lokale regelaars voor de deelnetwerken. In deze context kunnen de belangrijkste bijdragen van het proefschrift als volgt worden samengevat:

- We ontwikkelen een nieuw macroscopisch verkeersmodel voor stedelijke verkeersnetwerken, het zogenaamde BLX-model. Om de rekencomplexiteit verder te reduceren stellen we een macroscopisch model voor dat nog eenvoudiger is, het S-model. Het S-model is zeer snel, maar is ook nog steeds in staat om alle benodigde informatie te verschaffen die nodig is voor de regeling van verkeer. We introduceren ook een ander model met een aanpasbare bemonsteringstijd dat kenmerken van zowel het BLX-model als het S-model bevat en dat uitstekend geschikt is om de afweging tussen nauwkeurigheid en rekencomplexiteit in te stellen.
- We maken MPC-regelaars voor stedelijke verkeersnetwerken uitgaande van het BLX-model en het S-model. De online rekenefficiëntie van de MPC-regelaars gebaseerd op het S-model blijkt veel beter te zijn dan MPC gebaseerd op het BLX-model, en gaat slechts met een zeer beperkt verlies aan regelprestaties gepaard.
- Het niet-lineaire, niet-convexe optimalisatieprobleem van de MPC-regelaars voor stadsverkeer wordt voor het S-model geherformuleerd als een lineair optimalisatieprobleem met reële en gehele variabelen (*mixed integer linear programming*, MILP),

dat efficiënt opgelost kan worden door momenteel beschikbare MILP software. We stellen ook een verdere vereenvoudiging van het S-model voor, S\*-model genoemd, die resulteert in een kleiner MILP probleem. Zowel voor het S-model als het S\*-model geldt dat de online rekenefficiëntie van de MPC-regelaars sterk verbeterd wordt ten opzichte van de aanpak gebaseerd op niet-lineaire optimalisatie.

- Om zowel vertragingen als uitstoot en brandstofverbruik in grote steden te regelen, ontwikkelen we een geïntegreerd model voor verkeersstroom, uitstoot en brandstofverbruik. Op basis van dit model worden MPC-regelaars gemaakt die resulteren in een gebalanceerde afweging tussen het minimaliseren van de vertraging, de uitstoot en het brandstofverbruik.

S. Lin



# Summary

## Efficient model predictive control for large-scale urban traffic networks

Traffic congestion in urban areas, especially in big cities, is a challenging problem. One promising control methodology that can meet most of the needs for controlling and coordinating large-scale urban traffic networks using a variety of traffic control measures, is Model Predictive Control (MPC). MPC is an optimization-based control approach that can optimize and control multi-input multi-output systems with both linear and nonlinear constraints. MPC has several advantages for controlling large-scale traffic networks: MPC can easily coordinate various control measures implemented in traffic networks, it can predict the future traffic states to make long-term decisions, it is robust to disturbances and model uncertainty, and it can be implemented modularly. Therefore, the topic of this thesis is MPC for urban traffic networks.

However, a big difficulty to implement MPC in practice is the high on-line computational burden. When using MPC, at each time step, we have to solve an optimization problem within a limited period of time. If the optimization problem of MPC controller is too time-consuming to be solved on-line, due to the large scale of the optimization problem or due to the nonlinear, non-convex nature of the optimization problem, the MPC controller becomes real-time infeasible in practice, even though the problem is solvable in theory. Thus, the on-line computing efficiency of the MPC controllers is the key problem that has to be solved, before MPC can be applied to large-scale urban traffic networks.

Therefore, we present several ways in this thesis to address the computational problems arising when MPC is applied to control and coordinate large-scale urban traffic networks. The main methods considered in the thesis can be summarized as follows:

- **Model reduction:** Due to the nonlinear nature of the urban traffic models, the optimization problem of MPC controllers is a nonlinear non-convex optimization problem. It can be solved by nonlinear optimization algorithms, which requires a huge number of objective function evaluations based on the prediction model. Thus, the more complex the urban traffic model is, the longer time the on-line optimization process will take. Therefore, the on-line computational efficiency of the MPC controller can be improved by reducing the complexity of the prediction model of the MPC controller.

- **Reformulation of the MPC optimization problem:** Due to the nonlinear non-convex nature of the MPC optimization problem, the computational complexity will in practice increase almost exponentially when the scale of the controlled traffic network grows. Even though the on-line computational efficiency of the MPC controller can be improved by reducing the complexity of the prediction model of the MPC controller, the effect is still limited for huge traffic networks. Therefore, to avoid this, the on-line computational efficiency of the MPC controller can be also improved by reformulating the nonlinear non-convex MPC optimization problem into an optimization problem that can be solved much more efficiently.
- **Hierarchical control structure:** The computational complexity of a centralized MPC controller for a large-scale urban traffic network can be reduced by dividing the network into several smaller sub-networks, each of which results in a much lower computational burden. These traffic sub-networks have to be coordinated, so as to approximate the global control performance of a centralized MPC controller.

In this thesis, we have mainly investigated the first two approaches, i.e. “model reduction” and “reformulation of the optimization problem”. The on-line computing time of the MPC optimization problems was illustrated to be reduced by these two approaches for MPC controllers. Regarding to the third approach, we presented a general framework for MPC control of large-scale urban traffic networks, and analyze the different control structures for the network-wide traffic controller. All the centralized MPC controllers presented in this thesis are able to fit into the framework, and can act as local controllers for the urban traffic subnetworks. In this context, the main topics considered in the thesis can be summarized as follows:

- A new macroscopic traffic model for urban traffic networks, called BLX model, is established. To further reduce the computational complexity, a more simplified macroscopic model, the S model, is proposed. The S model is very fast, but is still able to provide all the necessary information that is needed for traffic control. We also introduce another model with an adjustable sample time step that contains features of both the BLX and the S model, and that is excellently suited to tune the trade-off between accuracy and computational complexity.
- MPC controllers are built for urban traffic networks based on the BLX and the S model. The on-line computational efficiency of the MPC controllers based on the S model is improved greatly compared to MPC based on the BLX model, and only a limited loss of control performance is incurred.
- The nonlinear non-convex optimization problem of the urban traffic MPC controllers is reformulated for the S model into a mixed-integer linear programming (MILP) problem, which can be solved very efficiently by existing MILP solvers. We also introduce a further simplification of the S model, called the S\* model, that results in a smaller MILP problem. For both the S model and the S\* model the on-line computational efficiency of the MPC controllers is further improved significantly compared to the nonlinear optimization approach.
- In order to control traffic delay as well as traffic emissions and fuel consumption in big cities, an integrated urban traffic flow, emissions, and fuel consumption model is



proposed. MPC controllers are built based on this model, which results in a balanced trade-off between minimizing travel time, reducing emissions, and fuel consumption.

S. Lin



# Curriculum Vitae

Shu Lin was born on March 17, 1981 in Chengde, Hebei Province, China. In 2003, she got the BSc degree from the department of automation in Qingdao University of Technology, China, and received excellent Bachelor dissertation award of the year.

In September 2003, she was enrolled in School of Control Science and Engineering, Shandong University, China, and worked on the project entitled as "Intelligent control algorithms in urban traffic control systems". In 2006, she received MSc degree in control science and engineering from Shandong University.

Since September 2006, she was enrolled in Department of Automation of Shanghai Jiao Tong University, Shanghai, China, as a PhD student supervised by Prof. Yugeng Xi, and worked on the project entitled as "Model predictive control for large urban traffic networks". During her PhD studying, in October 2008, she had the opportunity to visit Delft Center for Systems and Control, Delft University of Technology, The Netherlands, and then worked as a PhD student with her supervisors, Prof. Bart De Schutter and Prof. Hans Hellendoorn, continuing on the PhD research topic of "Efficient model predictive control for large-scale urban traffic networks" for two years.