Het verkeerskundig
laboratorium
voor studenten

ITS EDU LAB

Coordinated signal control for urban networks by using MFD

Maarten Strating

April 2010

Rijkswaterstaat

TU Delft

Ministerie van Verkeer en Waterstaat

Rijkswaterstaat

Coordinated signal control for urban traffic networks by using MPC

Maarten Strating

April 12, 2010

MSc Thesis Civil Engineering
Delft University of Technology

Colophon

Published by: ITS Edulab, Delft

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

Information: Henk Taale

Email: henk.taale@rws.nl

Author: Maarten Strating

Delft University of Technology

Master student Transport & Planning

Graduation committee:

Prof. Dr. Ir. S. P. Hoogendoorn
Committee chairman
Delft University of Technology
Faculty of Civil Engineering and Geosciences
Department of Transport & Planning

Dr. Ir. A. Hegyi
Delft University of Technology
Faculty of Civil Engineering and Geosciences
Department of Transport & Planning

R. Landman MSc
Delft University of Technology
Faculty of Civil Engineering and Geosciences
Department of Transport & Planning

S. Lin MSc
Delft University of Technology
Faculty of Mechanical, Maritime and Materials Engineering
Delft Center for Systems and Control

Dr. Ir. S. Hoogendoorn-Lanser
Rijkswaterstaat, Centre for Transport and Navigation

Dr. Ir. H. Taale
Rijkswaterstaat, Centre for Transport and Navigation

Ir. P. B. L. Wiggenraad
Delft University of Technology
Faculty of Civil Engineering and Geosciences
Department of Transport & Planning

Date: April 12, 2010

Status: Final

Voorwoord

In het kader van mijn afstuderen heb ik dit rapport geschreven. Het ITS Edulab is het samenwerkingsverband tussen de TU en de Dienst Verkeer en Scheepvaart (DVS). De meeste dagen van mijn afstuderen heb ik bij DVS doorgebracht, al heb ik de Chinese kamer op Werktuigbouwkunde ook vaak lastig gevallen. Verder waren heel wat uurtjes thuis achter de computer onvermijdelijk. De collega's van de afdeling Verkeersmanagement wil ik vanuit hier bedanken voor de gezelligheid, en niet te vergeten hun flexibiliteit en tomeloze inzet bij de potjes tafelvoetbal.

Wat de examencommissie betreft gaat mijn dank uit naar alle leden. Ieder heeft zo zijn eigen bijdrage aan dit proces gehad. Sascha, Paul en Ramon, dank jullie voor de kritische blik die jullie van tijd tot tijd over mijn werk hebben laten gaan. Andreas, als dagelijks begeleider heb ik veel aan je gehad. Dankzij je kritsche doch opbouwende houding heb je me de goede richting in geholpen. Ook bleek een uur praten veel meer kennis en vaardigheden op te leveren dan het lezen van een boek. Henk, dank voor je altijd snelle en bovenal flexibele computer ondersteuning. Zonder dat waren mijn simulaties nu nog niet halverwege geweest. Serge, jou waardeer ik omdat je altijd een leuk verhaal wilt delen en omdat je me in de beginfase hebt geholpen met het scheppen van kaders. Shu, I will miss the afternoons at 3ME. While working, it was always nice to talk with you and your roommates to find out about our cultural differences and personal issues.

Op de faculteit Civiele Techniek wil ik verder nog Maria Salomons en Peter Knoppers bedanken omdat ze altijd klaar stonden om mij van hunvoor mij aangepast software te voorzien.

Verder dank ik mijn familie en vrienden hartelijk, en en jou in het bijzonder Rika. Het zijn 9 mooie maanden geweest, maanden waarin het ook wel eens minder ging dan gehoopt. In voor- en tegenspoed heb ik jullie steun zeer gewaardeerd.

Den Haag, 12 april 2010

Summary

Dynamic Traffic Management (DTM) is preferred rather than the construction of new roads to increase traffic performance. This is because space is scarce and costs are high. Typical DTM measures implemented today are for example ramp metering, traffic signal installations, and variable message signs.

DTM measures are usually not coordinated and thus have no interaction with each other. It is possible that DTM measures solve a problem locally but create another problem elsewhere in the network. Then, those DTM measures should be coordinated in order to reach a better performance.

Coordination of all measures can be computationally expensive. To manage the scale, a hierarchical approach can be considered in which an upper level control calculates the desired traffic states in the subnetworks it is responsible for. Recently, empirical evidence is found that the traffic state of an urban road network can be described by a diagram with a constant shape: the Macroscopic Fundamental Diagram (MFD).

The MFD gives the relation between accumulation (number of vehicles in a network) and the weighted flow in the network. The flow is weighted according to the length of the link it is measured.

Main research questions

1. How to design a controller that controls DTM measures on a subnetwork level in a hierarchical setting?
2. What is the difference in performance between coordinated control and conventional control methods?
3. Is it possible to use MFD in a control concept for subnetworks?

The scope of the thesis is the control in a subnetwork by means of traffic light and ramp metering. Only motorized vehicles are considered. As a control method, Model Predictive Control is chosen because of its ability to predict, combine multiple objective functions, to deal with multivariate processes and to consider constraints.

The prediction model needed for MPC, is a macroscopic urban traffic model (S-model). The S-model performs calculations relatively fast, compared to a microscopic simulation model like Vissim or Paramics.

The MPC-controller can be linked to micro simulator Vissim. In this way, the calculated control can be applied in Vissim and the performance can be tested. MFD's can be generated as well.

Two types of experiments are performed:

1. MPC versus fixed time control and vehicle actuated control (Vrigen)
2. MPC (desired traffic state is imposed by an upper level control) versus MPC (no upper level control)

Two types of traffic load patterns will be applied:

1. Regular peak hour traffic
2. Event traffic

Before these experiments can be conducted, the compatibility between the S-model and Vissim need to be confirmed. In a sensitivity analysis it appears the compatibility is too low at this moment. The main source for this problem is the way link flow is modelled. For macroscopic traffic models, it is important that the time step is equal or smaller than the free flow travel time on the shortest link. A solution to this problem can be found in selecting a lower time step (higher computational time) or adjusting the model such that vehicles can pass more than 1 link in one time step.

The consequence of the incompatibility between the S-model and Vissim is that the MPC controller cannot be used for conducting experiments. Experiment 1 will only involve fixed time control and Vrigen control, while the second experiment is not executed. Note that the simulation environment for the MPC controller is finished and working technically.

The results for regular peak hour traffic are that fixed traffic control has constant MFD's and currently has the best performance. The Vrigen control method performs significantly worse. However the shape of the MFD's is not very different; the saturation point has a slightly lower maximum flow. The lower performance with respect to the fixed time controller is due to the use of offsets. In fixed time control, offsets can be defined and tuned while in Vrigen control offsets are random.

The results for event traffic are that the shapes of the MFD are similar for both fixed time control and Vrigen. Their performance is similar as well. The findings for these MFD's are limited, since no congested of saturation conditions were measured.

The conclusion is that in this thesis, the shape of MFD is found to be constant for fixed time control. When considering Vrigen control, some more variation was found, although the diagram still shows a typical fundamental diagram shape. These finding do not prove that the MFD is always constant when the control strategy is constant. It is therefore not yet suitable as a communication tool in a hierarchical traffic control concept. It is recommended to perform more experimental research to confirm whether there is one constant shape or if there are several constant shapes, or if there are conditions in which the shape of the MFD is unpredictable.

It is essential that the limitations in the S-model with respect to link flow modelling are solved in order to be able to use it for the MPC-controller. When this issue is solved, one can think about implementing other improvements. Another priority improvement is the support for offsets. This is essential to create coordination. The improvements will come at the price of computational power. So the choice for other additions to the S-model will be a trade-off between speed and accuracy.

Table of contents

1. INTRODUCTION 1

1.1 Introduction 1
1.2 Hierarchical control concept 2
1.3 Macroscopic Fundamental Diagram (MFD) 4
1.4 Controlling a subnetwork 7
1.5 Problem definition and research questions 7
1.6 Outline of the report 8

2. BACKGROUND INFORMATION 9

2.1 Introduction 9
2.2 Model Predictive Control (MPC) 9
2.3 Prediction Model (S-model) 12
2.4 Macroscopic Fundamental Diagrams (MFD) 12
2.5 Basic concepts in Traffic Control Engineering 14

3. RESEARCH APPROACH 17

3.1 Introduction 17
3.2 Hierarchical Control Concept 18
3.3 Formulation of the MPC controller 20
3.3.1 Objective formulation: Optimize TTS 21
3.3.2 Objective formulation: reach desired traffic state given by upper level control 21
3.3.3 Objective formulation: Control to a constant shaped MFD 23
3.3.4 Combined objectives 24
3.4 Study area 24
3.5 Description of the experiments 26
3.5.1 Traffic load pattern 28
3.5.2 Experiment 1 31
3.5.3 Experiment 2 31

4. SIMULATION ENVIRONMENT 34

4.1 Introduction 34
4.2 Overview of the simulation environment 34
4.3 Programming software (Matlab) 36
4.3.1 Different options 36
4.3.2 Working of the controller 36
4.4 Prediction model (S-Model) 38

4.4.1 Control horizon and prediction horizon 38
4.4.2 Network modelling 39
4.4.3 Phase modelling 42
4.4.4 Control variables 44
4.4.5 Objective function 45
4.4.6 Optimization of the objective function 45
4.5 VRIGEN CONTROL METHOD 48
4.6 FIXED TIME CONTROL 50
4.7 SOFTWARE USED BY TRAFFIC CONTROL METHODS 51
4.8 MICROSCOPIC SIMULATOR (VISSIM) 52
4.8.1 Different options 52
4.8.2 The network in Vissim 53
4.8.3 OD matrix 54
4.8.4 Random seed 55
4.8.5 Ramp metering 55
4.9 COMPARISON VISSIM AND S-MODEL 55
4.9.1 Control variables 56
4.9.2 Traffic load patterns 57
4.10 TRAFCOD AS A TRAFFIC CONTROLLER FOR MPC 59
4.11 NUMBER OF RUNS 60

5. RESULTS AND DISCUSSION 62

5.1 SENSITIVITY ANALYSIS BETWEEN S-MODEL AND VISSIM 62
5.1.1 Traffic load pattern 1 63
5.1.2 Traffic load pattern 2 71
5.1.3 Traffic load pattern 3 73
5.1.4 Traffic load pattern 4 75
5.1.5 Conclusion 77
5.2 EXPERIMENT 1: DIFFERENT CONTROL STRATEGIES COMPARED 77
5.2.1 Peak hour traffic pattern, fixed time control 78
5.2.2 Peak hour traffic pattern, Vrigen control 84
5.2.3 Event traffic pattern, fixed time control 87
5.2.4 Event traffic pattern, Vrigen control 88
5.2.5 Conclusion 90

6. CONCLUSIONS AND RECOMMENDATIONS 92

6.1 INTRODUCTION 92
6.2 CONTROLLER DESIGN IN A HIERARCHICAL CONCEPT 92
6.2.1 Cost function 92
6.2.2 S-model (prediction model) 92
6.2.3 Sensitivity analysis between S-model and Vissim 93
6.3 COORDINATED VERSUS CONVENTIONAL CONTROL 93
6.4 MFD AS A CONTROL CONCEPT IN HIERARCHICAL CONTROL 94
6.5 RECOMMENDATIONS 95

6.5.1 S-model *95*
6.5.2 Vissim *97*
6.5.3 Vrigen/Trafcod *98*
6.5.4 Hierarchical control *99*

BIBLIOGRAPHY 101

APPENDIX A 102

APPENDIX B 108

S-MODEL MANUAL 108
MICROSOFT VISUAL STUDIO 2005 108
QT3 LIBRARIES FOR WINDOWS 109
MATLAB 2008B 109
MICROSOFT VISUAL STUDIO 2005 110

APPENDIX C 112

NODE AND LINK NUMBERING IN S-MODEL 112

APPENDIX D 113

NUMBER OF RUNS, SAMPLE DATA 113

APPENDIX E 114

MFD'S PEAK HOUR 114

1. Introduction

1.1 Introduction

Every day, many urban and metropolitan areas in this world suffer from traffic congestion, both on urban roads and highways. Dynamic Traffic Management (DTM) is preferred rather than the construction of new roads to increase traffic performance. This is because space is scarce and construction costs are high. Typical DTM measures implemented nowadays are for example ramp metering, traffic signal controllers and variable message signs.

DTM measures are usually not coordinated and thus have no interaction with each other. It is possible that a DTM measure solves a problem locally but creates another problem elsewhere in the network. It is also possible that DTM measures have the potential to increase each other's effectiveness but fail to do so. In order to reach a better performance, DTM measures should be coordinated. The "Praktijk Proef Amsterdam" (PPA) aims at network-wide coordination of the different DTM measures in the region of Amsterdam.

The DTM measures considered in this thesis are traffic signal control and ramp metering, both for motorized vehicles.

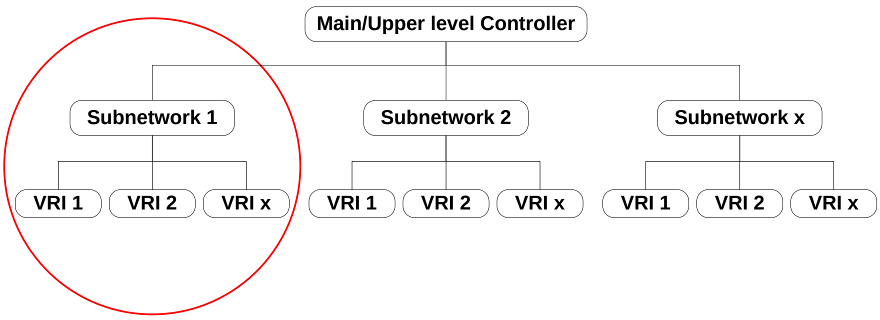


Figure 1.1 – Hierarchical control principle. The red circle indicates the scope of this thesis. VRI=Traffic Signal Control

The approach for network-wide coordination of DTM measures considered in this thesis is a hierarchical control principle (see Figure 1.1). In this control principle, a region's road network is divided into several parts, called subnetworks. The objective of the network-wide coordination is minimization of the total travel time. The main controller calculates a desired traffic state for each subnetwork in such a way, that the whole network will optimally perform. Each subnetwork will then coordinate all DTM measures in its area in such a way that it obeys to the main controller, and optimizes internal traffic performance as much as possible. The focus of this thesis is on the matter of subnetwork control, indicated with a red circle in Figure 1.1.

The choice for hierarchical approach in explained in Section 1.2. In Section 1.3, the Macroscopic Fundamental Diagram (MFD) is introduced as a way of communication between the main controller and the subnetworks. The approach for subnetwork control is explained in Section 1.4.

1.2 Hierarchical control concept

Coordination of all measures can be computationally expensive. The computational time will exponentially increase when the number of variables is larger. So, to reduce computational time, the number of variables should be lower.

Every DTM measure has at least one variable that has to be taken into account when coordination is performed. When looking at ramp metering, there's one variable. For traffic signal controllers, the number of variables depend on the control strategy and can be anywhere between 2 and 40, depending on the complexity of the controller. In case a coordination algorithm is predictive, the number of variables increases as the prediction horizon (the period of looking ahead in time) is longer. So the number of variables depends on the scale of the road network (the amount of DTM measures), the complexity of the DTM measures and the prediction horizon (if applicable).

Reducing the complexity of DTM measures means less control variables but it also means a reduced control freedom. Reducing the prediction horizon could mean a loss of accuracy or make it impossible to describe certain traffic phenomena. The last option to be considered is reducing the scale of the network. The scale of the network can be reduced in two ways:

- Consider only the main roads and other roads where congestion is likely to occur. This reduces the size of the network and a smaller size means fewer variables to consider in computations.
- Use a hierarchical control concept. In this concept, a city's road network is divided into several smaller subnetworks. See Figure 1.2 for an example of a possible break down of the road network in Amsterdam into several subnetworks. Those subnetworks have coordination within. The subnetworks on their turn are coordinated with respect to each by a higher level controller. In this way, coordination of the full road network is

achieved. The principle of hierarchical control is given in Figure 1.1.

The first three simplifications (reduce the complexity of DTM measures, reduce prediction horizon, only consider roads where congestion is likely to occur) could be considered to reduce computation time. They involve a trade-off between accuracy and computational speed. The hierarchical approach on the other hand (the method of dividing a city's road network into several smaller subnetworks) could make it possible to maintain accuracy while reducing computational time substantially.

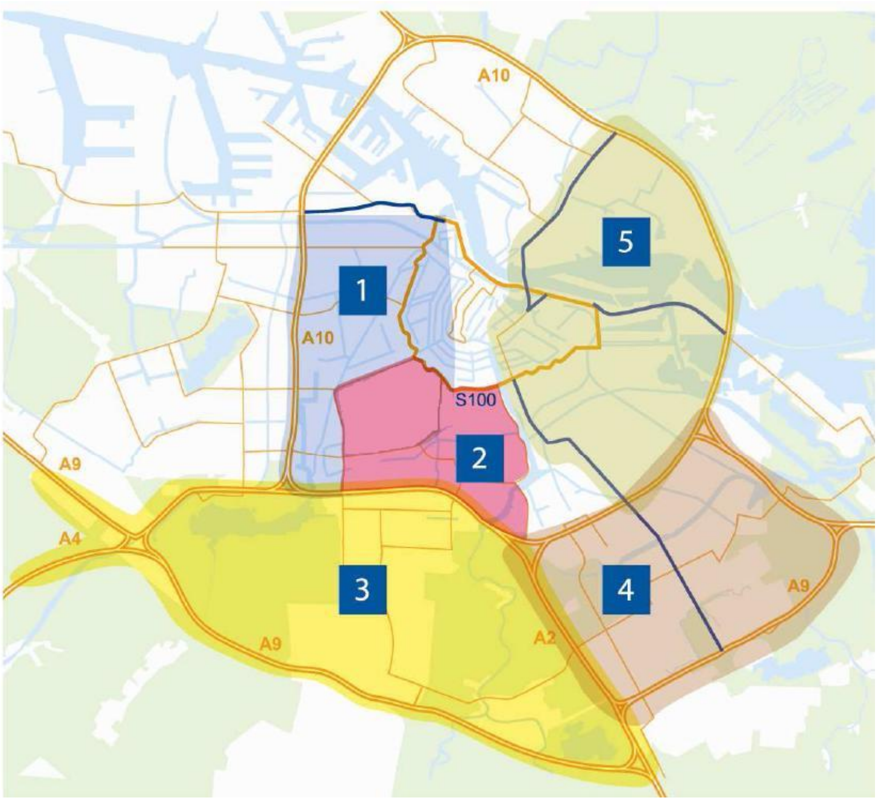


Figure 1.2 - Subnetworks in the Amsterdam road network (Rijkswaterstaat, 2009)

The following example will show why the hierarchical control approach has the highest potential to reduce computational time with respect to the other three strategies discussed. Assume that the total network involves 100 variables to be coordinated and the computational time is given by the relation $y = 1{,}1^x$, in which x is the number of variables and y is the computation time (see Figure 1.3). When this network is divided into five equal parts (with 20 variables), the computational time for each subnetwork is 0.049% of the total network's computational time: 13781 versus 6.7 seconds. For all subnetworks, the total computational time adds up to 0.244% with respect to that of a full network. A necessary condition to achieve coordination for the full network is that it is possible to coordinate subnetworks at reasonable computational expenses by a higher level controller. If that computational time is less than 99.76% (with respect to the case of full coordination), there is already an improvement. For real-time applications however, a computational time in the same order as found for a subnetwork in this example, is necessary. When comparing the different options to the objective of reducing computational time, it is clear from this example that the hierarchical control concept (Figure 1.1) can be expected to deliver the highest reduction in computational time.

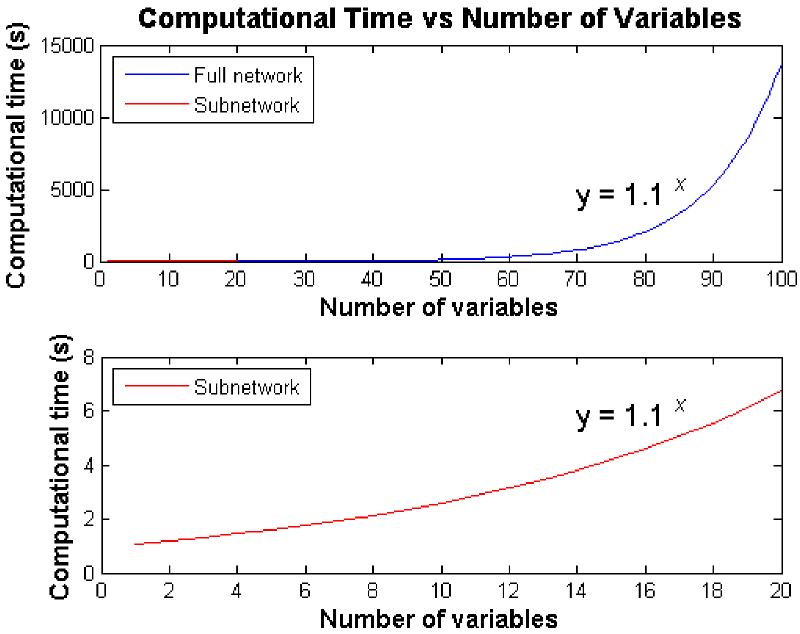


Figure 1.3 – The influence of number of variables (20 versus 100) on the computational time

The control concept for the main controller should be as simple as possible to keep the computational time low. In order to keep the hierarchical control concept simple, the communication between the main control and the subnetworks should be as simple as possible. Note that there is no communication between the subnetworks. The concept of Macroscopic Fundamental Diagram (MFD) could be useful to provide the (simple) communication between subnetworks and a higher level controller. The concept of MFD will be explained in Section 1.3.

1.3 Macroscopic Fundamental Diagram (MFD)

Recently, empirical evidence is found that the traffic state of an urban road network can be described by a diagram (Figure 1.4) with a constant nature: the MFD (Geroliminis and Daganzo, 2009). The MFD gives the relation between accumulation (the amount of vehicles present in the network) and the traffic flow in the network. The traffic flow consists of traffic flows inside the network, traffic flows going entering the network and traffic flows leaving of the network.

Figure 1.4 – Example of a Macroscopic Fundamental Diagram (MFD)

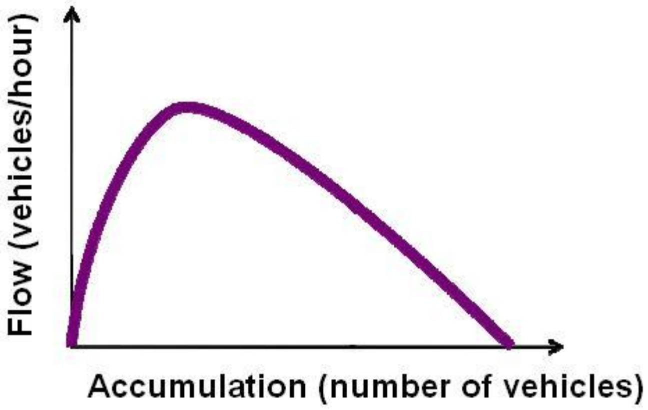
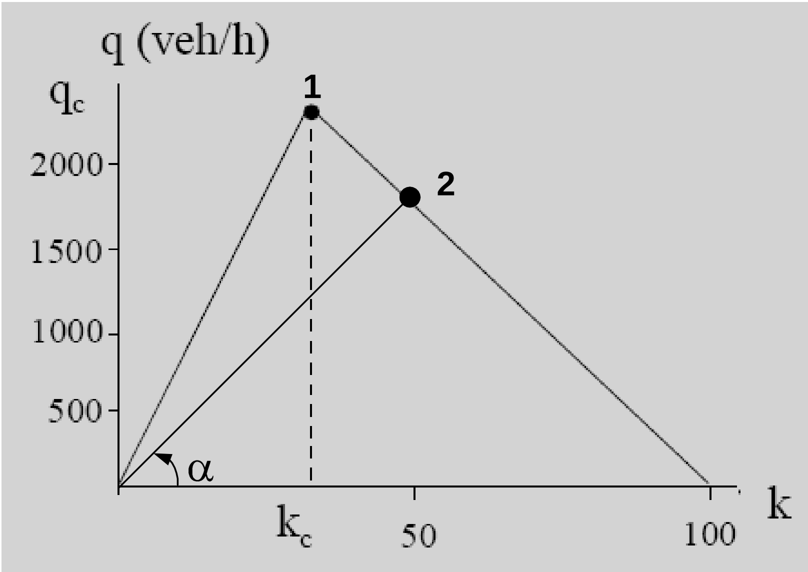


Figure 1.5 – Example of a Fundamental Diagram (FD), giving the relation between density (k) and flow (q) at a cross section of a road. The speed is indicated by the angle α. (Hoogendoorn, 2000)



The interpretation of the MFD (Figure 1.4) is similar to the commonly used Fundamental Diagram (FD) (Figure 1.5) in traffic science; as the accumulation (MFD) or density (FD) is increasing, the traffic flow increases until the point where congestion is reached. In Figure 1.5, point 1 indicates the maximum flow where point 2 indicates a state of congestion. The difference is that a regular FD describes the traffic state (speed, density and flow) at the cross section of a road, while the MFD gives the traffic state (average speed, average density and flow) in a road network. The shape of the MFD itself is a result of the lay-out and infrastructure of the network. One can think of flyovers, signal timers and dedicated bus and taxi lanes. Every network has its own unique MFD and these factors (as well as many others) are expected to influence its shape (Geroliminis and Daganzo, 2008). While these factors are constant, the shape of the MFD is expected to remain constant as well. The traffic state of the network can then be described by one point on the MFD.

In a hierarchical traffic control concept, the MFD could play a role in communication between the upper level controller and the subnetworks. The principle of hierarchical control is given in Figure 1.1 and communication by using MFD is shown in Figure 1.6. The upper level controller will have access to relevant real-time traffic data, such as the actual traffic states in all subnetworks. Also more detailed information like current traffic flows on specific links could be relevant data. From this data, it calculates the desired traffic states in all subnetworks. The control method of the upper level controller is not relevant in this thesis. After calculating the desired traffic state for all subnetworks, the traffic states can be communicated to each subnetwork as a point on their respective MFD's. It is then up to each individual subnetwork to optimize its traffic performance while respecting the desired state given by the higher level control as much as possible. In Figure 1.6, both the desired and the actual state are randomly chosen as an example.

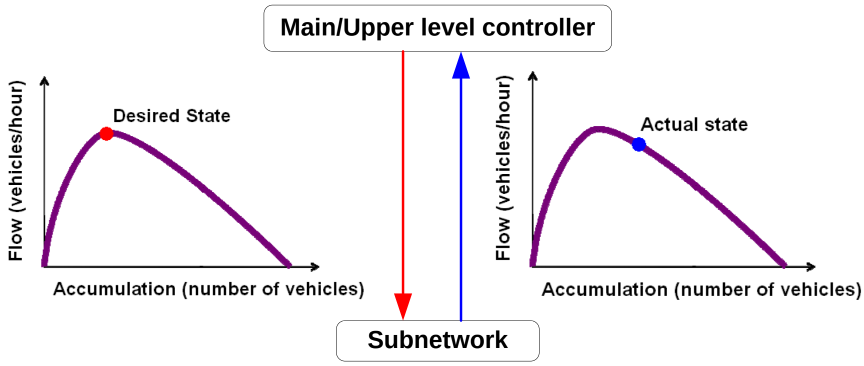


Figure 1.6 – Communication between the main controller (sends a desired state) and a subnetwork (sends its actual state). The states in this picture are chosen randomly.

To use MFD to communicate the traffic states, it is important that the shape of the MFD is constant. Otherwise, the main control will calculate a state for a subnetwork that is not valid. The result could be for example that a certain flow is desired, but in reality the flow is lower. Then, traffic jams could be created while that is not the intention.

At this moment, it is not clear under which conditions the MFD will have a constant shape. Finding out to which extent the MFD will be constant is part of this thesis.

The way a subnetwork can be controlled by using the input from the upper level controller (by means of MFD), is described in Section 1.4.

1.4 Controlling a subnetwork

The objective of the upper level controller is to minimize the total time spent (TTS) on a network-wide level. In order to achieve this goal, the control of a subnetwork should be subject to two objectives:

- reach a desired traffic state given by the main control
- optimize internal traffic performance (TTS)

These two objectives can be complementary but can be conflicting as well. The global objective is minimizing TTS in the whole network. This objective has a utilitarian[1] nature; it is possible that optimal performance in the whole network is achieved at the cost of congestion in one subnetwork. This subnetwork could have performed better in case no coordination between the subnetworks was present.

The control concept for a subnetwork should be able to deal with conflicting objectives. Since both traffic signal control and ramp metering are present, the control concept should be capable of controlling multiple measures simultaneously. Model Predictive Control (MPC) is a control method that meets these demands, therefore MPC is chosen as the control method on subnetwork level. A general description of MPC is given in Section 2.2.

1.5 Problem definition and research questions

Summarizing the previous sections, the problem definition can be formulated as follows:

Problem definition

Uncoordinated Dynamic Traffic Management (DTM) measures work in an isolated manner and fail to support each other or they may work counterproductive. At this moment, DTM measures are operational without coordination.
A proposed means to achieve coordination is the use of a hierarchical control concept. On subnetwork level, MPC seems a suitable control method. For the communication between upper level control and subnetwork control, the use of MFD looks promising but it is uncertain whether the use of MFD is viable in this control concept.

Three research questions can be formulated:

[1] Utilitarism: The greatest good for the greatest number of people

Main research questions

1. How to design a controller that controls DTM measures on a subnetwork level in a hierarchical setting?
2. What is the difference in performance between coordinated control and conventional control methods?
3. Is it possible to use MFD in a control concept for subnetworks?

The scientific contribution of this thesis is in answering the research questions. A control concept for subnetwork control will be presented, which functions in a hierarchical setting. This control concept will be compared to conventional control concepts. In order to keep the hierarchical control concept simple, communication between upper level control and subnetwork control should be as simple as possible. To use MFD as a communication concept, its dynamics should be clear; does the MFD have a constant shape? If yes, under which conditions? If not, what should be done to be able to use MFD in control concept for subnetworks? These questions will be answered in this thesis.

The technical contribution of this thesis is about building an MPC controller that is able to control a subnetwork traffic signal control and take into account ramp metering. An MPC controller consists of three different parts of software (see Section 2.2) which needed to be integrated in order to work. To test this controller, it should be implemented in reality or in a virtual reality. The first option is not feasible, so traffic simulation software is used. The MPC controller and the traffic simulation software have been integrated. The software integration is done in such a way that the different parts of the software are easy to replace or improve. This allows future research with a focus on one of the elements in hierarchical control, rather than focussing on the integration of all elements. Software integration is necessary in order for the software to work flawlessly and can be rather time consuming.

1.6 Outline of the report

Chapter 2 contains background information about MPC, MFD and some specific term used in traffic control engineering. Chapter 3 describes the research approach and the kind of experiments that will have to be performed in order to get answers to the research questions. In Chapter 4, the approach with respect to the software for performing the experiments is explained. Chapter 5 contains the results and interpretation of the results. Chapter 6 gives conclusions and recommendations with discussion for further research. The appendices contain detailed information about the software used.

2. Background Information

2.1 Introduction

This chapter is intended to inform the reader about the following subjects:

- Model Predictive Control (MPC), Section 2.2
- Prediction model to be used in MPC (S-model), Section 2.3
- Macroscopic Fundamental Diagram (MFD), Section 2.4
- Basic terminology used in traffic control engineering, Section 2.5

The reader can omit the sections on subjects that she or he is already familiar with.

2.2 Model Predictive Control (MPC)

MPC is a control method which has the following main advantages:

- The ability to handle multiple control inputs
- Combine multiple objective functions
- (Nonlinear) constraints are explicitly considered

A consequence of the predictive nature of MPC is the possibility to perform temporary suboptimal in order to reach an optimal overall performance. Today's main application of MPC is in the control of chemical plants.

The general structure of an MPC-controller is given in Figure 2.1. First, there is a process. In this example it is a plant, but this can be any controllable process. The symbols used are listed in Table 2.1. The interaction between the process and the MPC-controller can be described by the following six steps.

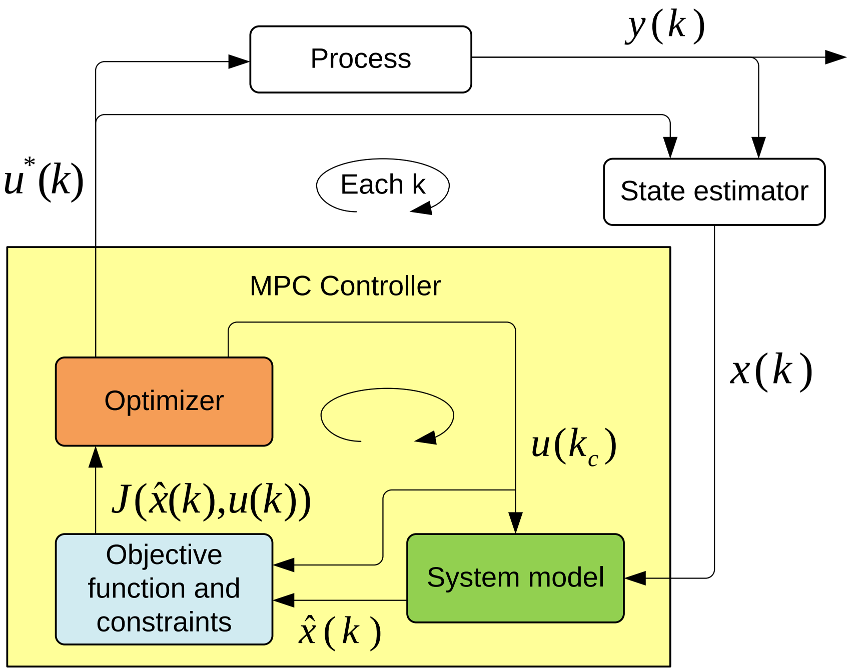


Figure 2.1 – Structure of an MPC controller

Table 2.1 – Symbols used to describe MPC

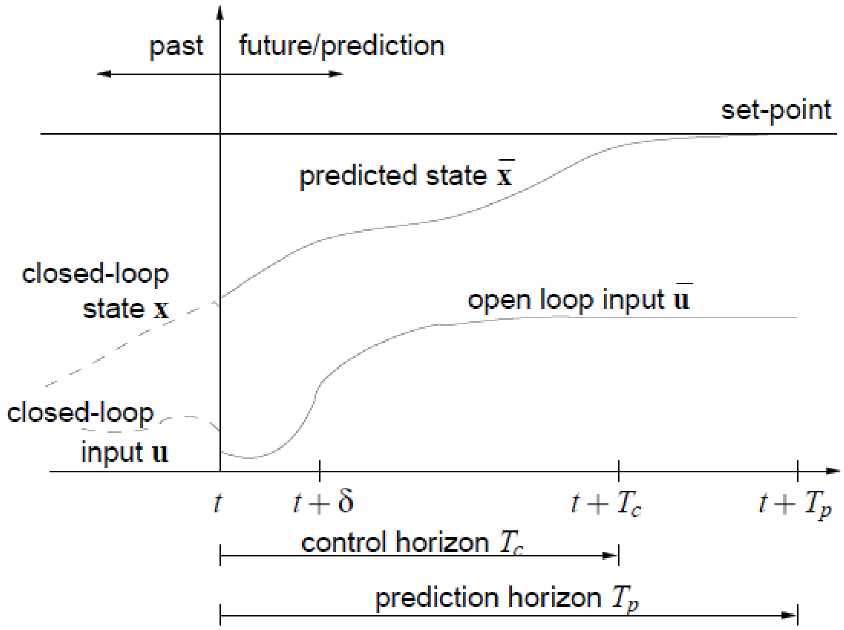
Symbol	Meaning
k	Time index
δ	Control time
T_c	Control horizon
T_p	Prediction horizon
$x(k)$	Estimated state of the process at time step k
$\hat{x}(k)$	Predicted states at time step k for T_p
$y(k)$	Output of the process at time step k
$u(k)$	Control signal at time step k for T_p
$u^*(k)$	Part of the optimized control signal (with duration δ) that will be used in the process at time step k
$J(\hat{x}(k),u(k))$	Objective function

1. The process in controlled by some control input $u^*(k)$. The output of the process is called $y(k)$.
2. Based on both the control signal $u^*(k)$ and process output $y(k)$, the state of the process is estimated in case this cannot be measured in a direct or accurate way). Otherwise the state can be directly measured. The (estimated) state is called $x(k)$.
3. The (estimated) state $x(k)$ is read by the system model. This model calculates the expected state of the system in the future

$\hat{x}(k)$, based on the prediction horizon and input (control) variable $u(k)$.

4. An optimal control signal $u^*(k)$ is computed by the optimizer. The optimizer minimizes the objective function $J(\hat{x}(k),u(k))$, while respecting the constraints on the control signal. This is an iterative process. The amount of iterations depends on the desired accuracy and convergence criterion.
5. The first part of the calculated optimal control signal $u^*(k)$ is implemented in the process for time δ (see Figure 2.2).
6. Continue with step 2.

Figure 2.2 – Control horizon and prediction horizon (Findeisen and Allgöwer, 2002)



When making a prediction with the prediction model, there is a control horizon (T_c) and prediction horizon (T_p), see Figure 2.2. The prediction horizon is the time for which the process is predicted. The objective function $J(\hat{x}(k),u(k))$ is always computed for T_p. The control horizon is the time in which the control input $u(k)$ can be varied when optimizing. During the time (T_p-T_c), the control input $u(k)$ is taken constant. The control horizon can never be larger than the prediction horizon; at most they can have equal values. There are two main reasons to choose a control horizon smaller than the prediction horizon:

1. save computational time; fewer variables have to be optimized.
2. force the controller to control in the start of the process. There is a danger that the controller will postpone a change in $u(k)$ beyond the control time δ. In the next prediction, the same can happen and still no change in $u(k)$ has occurred. In this way, nothing will happen so that is why the controller is forced to act at the start of the process.

2.3 Prediction Model (S-model)

When one wants to apply MPC to urban traffic control, in order to optimize the control to a certain objective function, the prediction model should be capable of predicting urban traffic. Such a model is the S-model designed by Shu Lin. The description of this model is directly taken from Lin et al, 2009 and can be found in Appendix A.

2.4 Macroscopic Fundamental Diagrams (MFD)

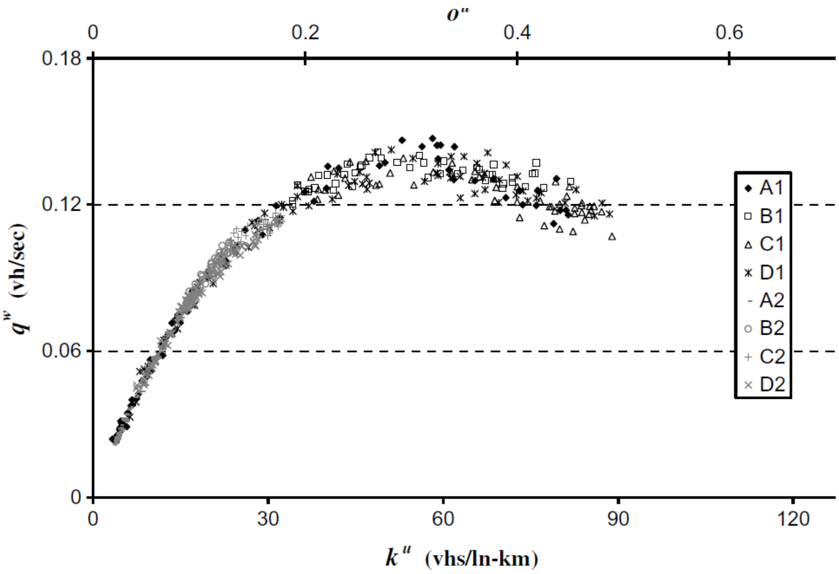


Figure 2.3 – MFD, Weighted average flow vs. density (Geroliminis and Daganzo, 2008)

In the past 40 years, various theories were proposed to describe vehicular traffic on an aggregate level (Geroliminis and Daganzo, 2008). One of them is a reproducible Macroscopic Fundamental Diagram (MFD), relating the accumulation (the number of vehicles to the average flow. The idea of an MFD is rather old (Godfrey, 1969). The verification of its existence however, is recently found in a field test in Yokohama.

An example of a MFD can be seen in Figure 2.3. The units of the x- and y-axis are density (k^u) and weighted flow (q^w) respectively. They are defined as follows:

$$q^w = \frac{\sum_i q_i l_i}{\sum_i l_i} \quad \text{Eq. 2.1}$$

$$k^u = \frac{o^u}{s} = \frac{1}{s}\frac{\sum_i o_i}{\sum_i l_i} \quad \text{Eq. 2.2}$$

In which i denotes the number of links, l_i is the link length for link i, o_i is the occupancy at link i and s denotes the mean vehicle length (s=5,5m, Geroliminis and Daganzo, 2008).

Data to produce an experimental MFD was collected in Yokohama (Japan). The area of analysis has a triangular shape (see Figure 2.4) and has a surface area of about 15 km^2. Two types of detectors were used to collect the data:

- Fixed sensors; 500 ultrasonic and loop detectors, placed 100 m upstream of most major intersections.
- Mobile sensors; 140 taxis were equipped with a GPS device and a data logger.

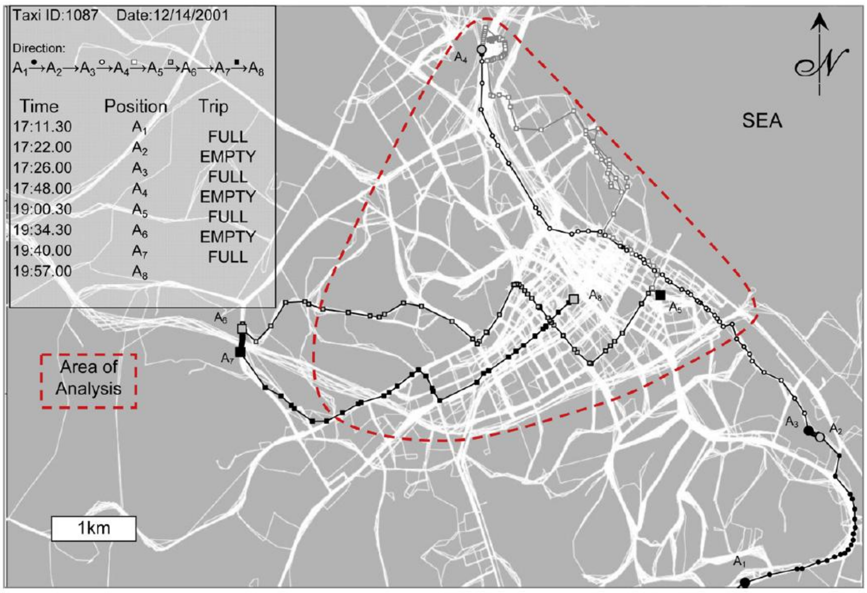


Figure 2.4 – Research area in Yokohama (Geroliminis and Daganzo, 2008)

The result of the research is that urban areas in the order of 10km^2 should have a well-defined MFD, independent of the traffic demand. "..the amount of street space allocated to cars and busses, street closures, flyover construction or new signal timings surely affect a neighbourhood's MFD. Therefore we are currently studying how a city's MFD depends on its infrastructure" (Geroliminis and Daganzo, 2008).

This outcome looks promising, as a constant MFD allows simple communication in the hierarchical control concept described in Section 1.2. From this thesis, the information on signal timings is important. From this MFD research, it is not clear whether the MFD will change shape if one changes the control strategy, or change some timers while keeping the same control strategy. Under which condition an MFD changes shape is relevant for the hierarchical control concept. It has to be found out to which extent the MFD will be constant.

2.5 Basic concepts in Traffic Control Engineering

The following concepts in signalized traffic control will be explained in this section:

- Cycle time
- Phases and phase schemes
- Clearance time

These concepts will play a role in the controller design, so it is important for the reader to be familiar with these concepts. The theory is derived from Van Zuylen et al, 2009.

Cycle time

Every controlled intersection has a cycle time. The cycle time is defined as the time in which all phases have occurred once.

Phases and phase schemes

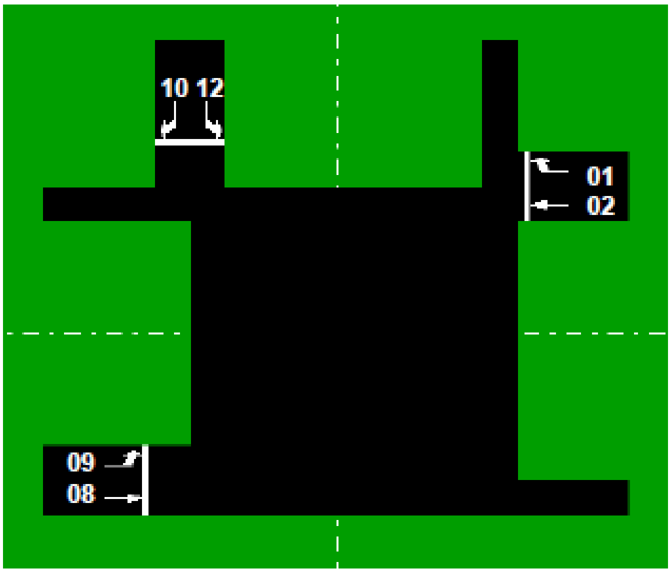


Figure 2.5 – Signal group codes at a T-junction (Van Zuylen et al, 2009)

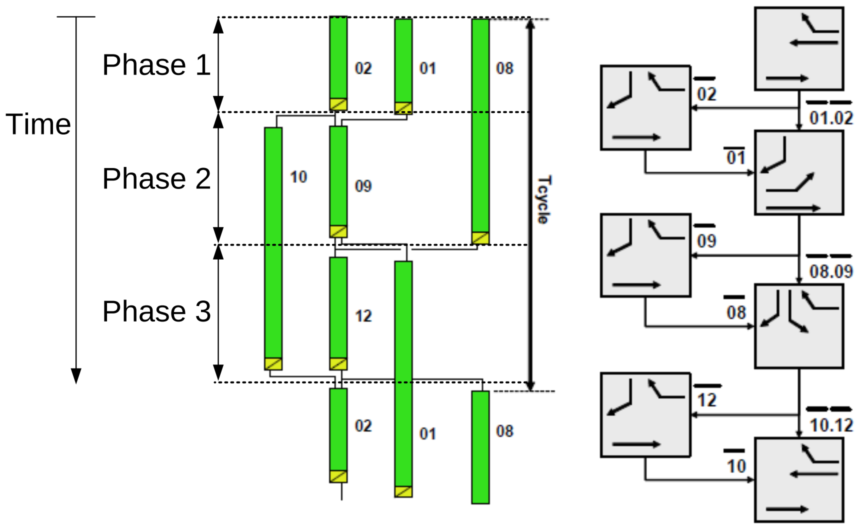
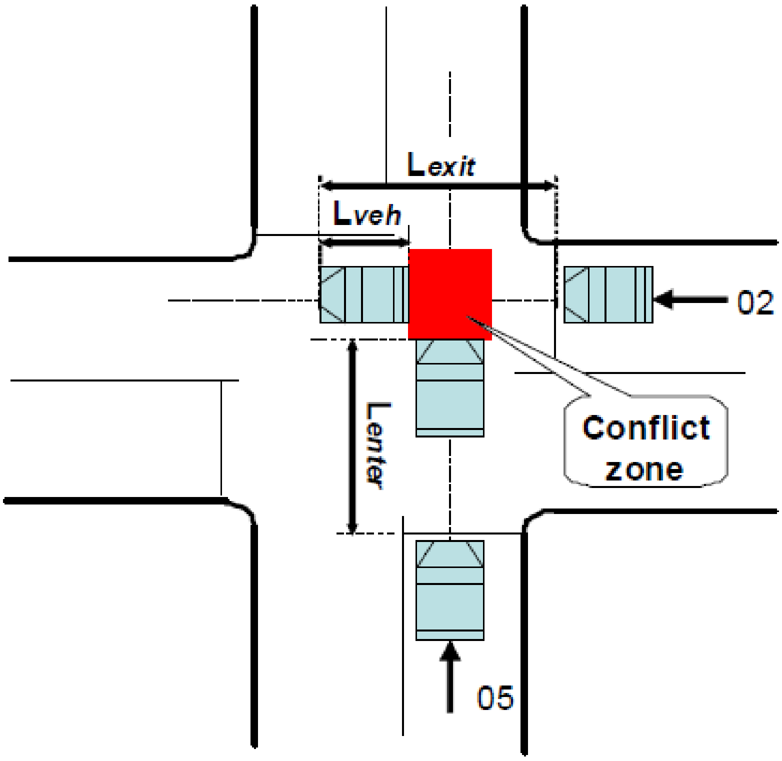


Figure 2.6 – Phases and cycle time (Van Zuylen et al, 2009)

When considering a T-junction for example as in Figure 2.5, there are six signal groups (1, 2, 8, 9, 10 and 12). The critical conflict groups consist of signal groups 2, 9 and 12. The other groups (1, 8 and 10) can be realized in multiple phases (see Figure 2.6). A more flexible approach is shown at the right hand side of Figure 2.6. For example, when there is no demand for signal group 2, signal group 10 can turn green.

Clearance time

Figure 2.7 – Conflict zone at an intersection (Van Zuylen et al, 2009)



An intersection is signalized to prevent collisions of vehicles at conflict zones. An example of a conflict zone can be found in Figure 2.7. It is considered safe to give a green signal after all vehicles from conflicting streams are outside the conflict area. The clearance time is defined as the time necessary to just clear the conflict zone. A shorter clearance time would be unsafe, while a longer clearance time would increase the internal lost time of the intersection.

From the example in Figure 2.7, it can be seen that the distance from the stop line to the conflict zone is shorter for signal group 2 when compared to signal group 5. Therefore, the clearance time for this conflict is lower for signal group 2. To minimize the internal lost time of the intersection, the sequence 2-5 is better when compared to 5-2. The result could be a lower cycle time, or longer green times. The decrease in cycle time is in the order of several seconds. It is likely that the cycle time will be shorter. Depending on the other conflict groups and the traffic flows, it could also be that the sequence 5-2 is better.

3.Research Approach

3.1 Introduction

This chapter explains the approach to answer the main research questions. For convenience, they are repeated:

Main research questions

1. How to design a controller that controls DTM measures on a subnetwork level in a hierarchical setting?
2. What is the difference in performance between coordinated control and conventional control methods?
3. Is it possible to use MFD in a control concept for subnetworks?

The control method that will be used to control DTM measures on a subnetwork level is Model Predictive Control (MPC). Its design consists of three parts: the system model, the objective function and the optimizer. The system model is the S-model, which is explained in (Appendix A). The formulation of the objective function will be given in Section 3.3. Building an optimizer is an art in its own, so an existing optimizer will be chosen. This depends on the software that will be used.

The MPC controller will function in a simulation environment. A field test is not feasible due to relatively high costs involved when compared to the use of a computer. The description of the simulation environment is part of the research approach. However, since the simulation environment is so extensive, it will be described in Chapter 4.

The hierarchical control concept is described in Section 3.2. The formulation of the objective function is given in Section 3.3. The study area is described in Section 0. A description of the experiments that will be conducted can be found in Section 3.5.

3.2 Hierarchical Control Concept

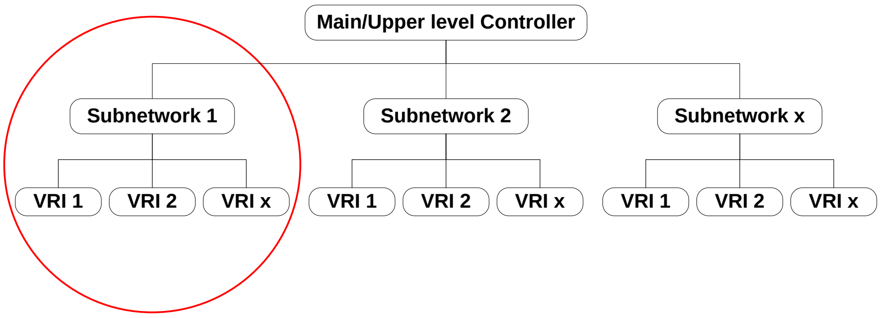


Figure 3.1 – Hierarchical control principle. The red circle indicates the scope of this thesis. VRI=Traffic Signal Control

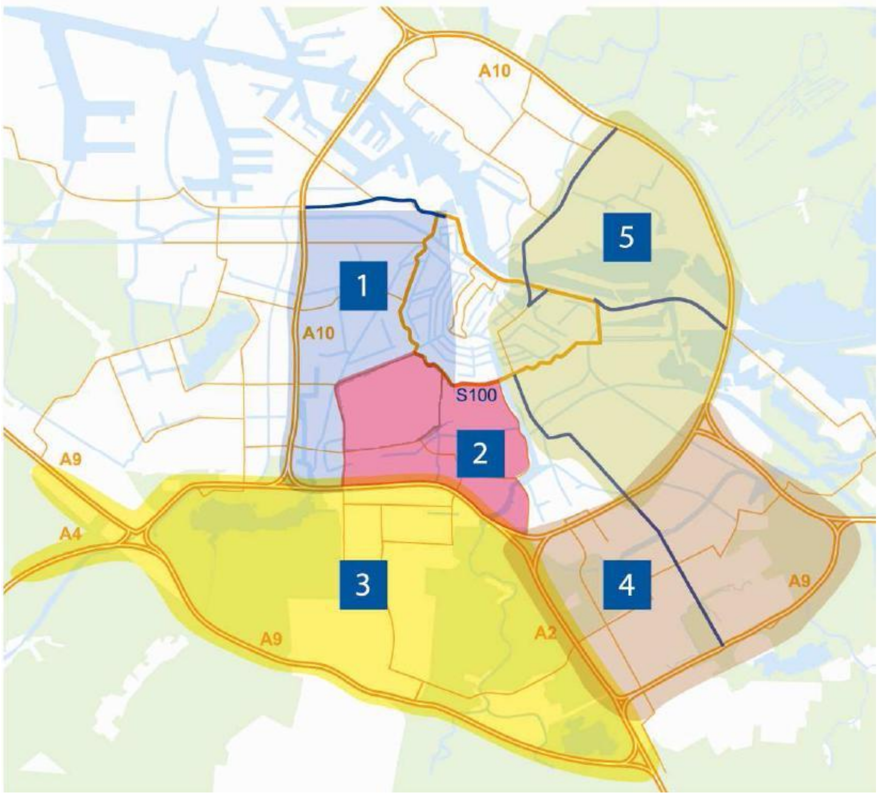


Figure 3.2 - Subnetworks in the Amsterdam road network (Rijkswaterstaat, 2009)

The approach for network-wide coordination of DTM measures considered in this thesis is a hierarchical control principle (see Figure 3.1). In this control principle, a region's road network is divided into several parts called subnetworks (see Figure 3.2 for an example). The objective of the network-wide coordination is minimization of the total travel time. In Section 1.2 it was motivated with an example why this hierarchical control concept has a high potential to save computational time with respect to network wide coordination on one level. In terms of hierarchical control, coordination on one level means no main controller, there is just one network. Its control method is comparable to the control method suggested for subnetworks in this thesis.

The main controller calculates a desired traffic state for each subnetwork in such a way, that the whole network will optimally perform. In theory, this can be any point present on a subnetwork's MFD. The calculated traffic state consists of a global traffic state which can be indicated on the subnetwork-specific MFD and additional (more specific) information on desired flows at the edge of a subnetwork. The control method used by the main controller is not discussed; in this thesis possible control actions taken by the main controller are assumed. However, a general remark can be made: the control concept for the Main controller should be as simple as possible to keep the computational time low.

Each subnetwork will coordinate all DTM measures in its area in such a way that it obeys to the main controller, and optimizes internal traffic performance as much as possible. In other words, to achieve the goal of network-wide minimization of total travel time, the control of a subnetwork should be subject to two objectives:

- optimize internal traffic performance (TTS)
- reach a desired traffic state given by the main control

From a traffic engineering point of view, it can be expected that these objectives result in optimal internal traffic performance in the subnetwork when the main control has no desired traffic state. If there is a desired state imposed by the main controller, this could cause restrictions with respect to control freedom for the subnetwork. It could mean for example that only a limited amount of vehicles can leave the subnetwork. Within this limitation, the subnetwork will minimize total travel time with the control freedom that is still left.

Note that also other objectives can be chosen, if the prediction model is capable of calculating the necessary states. One can think for example of minimizing the number of stops and minimizing the total emissions. The prediction model used in this thesis (S-model) is designed to calculate TTS and estimate the number of vehicles moving through the network. The S-model was not designed to provide other output, so that is why no other objectives are chosen. Furthermore, TTS is a good indicator of traffic performance in a network.

The two objectives considered by the subnetwork controller can be complementary but they can be conflicting as well. It is possible that optimal performance in the whole network is achieved at the cost of congestion in one subnetwork. This subnetwork could have performed better in case no coordination between the subnetworks was present. In the coordinated case, that subnetwork will “suffer” (experience congestion) so the rest of the network can benefit (no or less heavy congestion relative to a situation without coordination).

In order to keep the hierarchical control concept simple, the communication between the main control and the subnetworks should be as simple as possible. Note that there is no communication between the subnetworks (Figure 3.1). The concept of Macroscopic Fundamental Diagram (MFD) could be useful to provide the (simple) communication between subnetworks and a higher level controller.

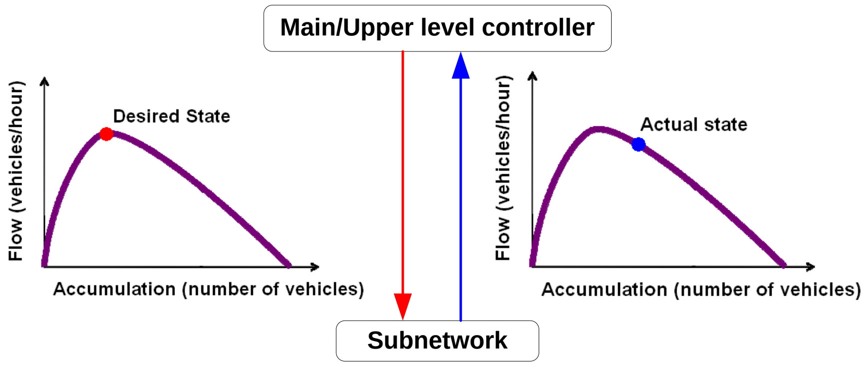


Figure 3.3 – Communication between the main controller (sends a desired state) and a subnetwork (sends its actual state). The states in this picture are chosen randomly.

When the shape of the MFD is constant, it can be used to communicate traffic states in both ways between the main controller and the subnetwork controller. The traffic states in the subnetworks are a necessary input for the main controller to calculate a control signal. They can communicate their state by a point on the MFD and additional flow information on relevant links. The control signal calculated by the main controller is communicated to each subnetwork, also by means of the same kind of information: a point on the MFD that indicates the desired global traffic state and desired flows at specific links. These will be only incoming and outgoing links. The traffic distribution inside the subnetwork is controlled by the subnetwork itself and is not relevant for the main controller.

3.3 Formulation of the MPC controller

The objective of the upper level controller is to minimize the total time spent (TTS) on a network-wide level. In order to achieve this goal, the control of a subnetwork should be subject to two objectives:

- optimize internal traffic performance (TTS)
- reach a desired traffic state given by the main control

The formulation for optimizing TTS is given in Section 3.3.1. The formulation of reaching a desired state can be done in two ways. The first formulation is for the case where the shape of the MFD is being investigated, rather than being known (Section 3.3.2). Hence, the MFD is not part of the formulation for the objective function. Secondly, a formulation will be given for the case where the MFD is not constant, and the MPC controller will control in such a way, that the shape of the MFD will remain constant (Section 3.3.3). The time available for executing this thesis was too short for testing this MPC formulation. It is rather an encouragement for further research. In Section 3.3.4 the complete MPC formulation is given.

3.3.1 Objective formulation: Optimize TTS

The objective function is denoted by J. The only performance indicator for TTS is the total travel time from all vehicles in the network. The total simulation time can be divided in l cycle times and the network can be divided in m links. The travel time of all vehicles in link m during cycle l is denoted by $ts_{i,j}$. Mathematically this can be described as follows:

$$\min(J_{TTS}) \qquad \text{Eq. 3.1}$$

$$J_{TTS} = \alpha_1 \sum_{i=1}^{l} \sum_{j=1}^{m} ts_{i,j} \qquad \text{Eq. 3.2}$$

The coefficient α_i will later be used to balance the different terms in the final objective function.

3.3.2 Objective formulation: reach desired traffic state given by upper level control

A desired traffic state will mean in practice that the upper level controller wants to limit the outflow of the subnetwork or ensure a minimal inflow of traffic from an adjacent subnetwork. In traffic control, a limited outflow means that ramp metering is applied for traffic entering a motorway or that limited green light time is given to traffic that wants leave the subnetwork by an urban road. A minimal inflow means that a minimal amount of green light is desired for incoming traffic (for example to prevent traffic jam at an off ramp).

Mathematically, a desired traffic state can be formulated as a reference flow that should be achieved. If not, a penalty should be applied and this penalty should be higher as the deviation from the reference value is higher. An appropriate way to describe this penalty mathematically is by a quadratic function, because a high deviation gets a relatively heavy penalty when compared to a small difference.

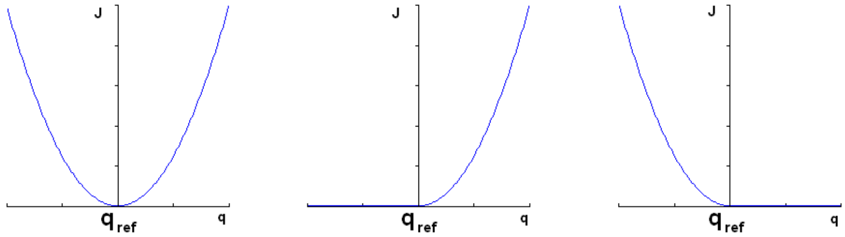
The function looks as follows: $J = (q - q_{ref})^2$ (see Figure 3.4, graph 1).

This quadratic formulation is suitable for both a desired maximum outflow and a desired minimal inflow. The second possibility is to have separate functions for desired maximum outflow (Figure 3.4, graph 2) and desired minimum inflow (Figure 3.4, graph 3) respectively:

$J = (q - q_{ref})^2$, for $(q - q_{ref}) \geq 0$, otherwise $J = 0$ (graph 2)

$J = (q - q_{ref})^2$, for $(q - q_{ref}) \leq 0$, otherwise $J = 0$ (graph 3)

Figure 3.4 – Possible objective functions for a desired flow



The second possibility (graph 2 and 3 in Figure 3.4) has the apparent advantage of not giving a penalty when the realized flow is less than the desired flow in case of maximum outflow. In the cases of minimum inflow, no penalty will be given if a higher traffic flow enters the network with respect to what was desired.

For the first formulation (Figure 3.4, graph 1), it means in situations where the reference flow cannot be met, unnecessary penalties will be imposed. Consider the following example. There is a desired maximum outflow of 600 vehicles per hour, while only 500 vehicles per hour are approaching. No matter what the traffic control will do, in all cases at least the same penalty will be applied. The result is that the absolute value of the objective function is higher. This is not a problem since the calculated control signal will be no different when compared to a situation where no penalty is involved.

Therefore, the straightforward choice is made to propose the following objective function. The cycle index is denoted by l, m is the link index, n is the amount of outflows with a desired reference outflow and r the amount of inflows with a desired reference inflow:

$$\min(J_{Qref})$$ **Eq. 3.3**

$$J_{Qref} = \alpha_2 J_{Qref,out} + \alpha_3 J_{Qref,in}$$ **Eq. 3.4**

$$J_{Qref,out} = \sum_{i=1}^{l} \sum_{k=1}^{n} (q_{i,k} - q_{ref,i,k})^2$$ **Eq. 3.5**

$$J_{Qref,in} = \sum_{i=1}^{I} \sum_{p=1}^{r} (q_{i,p} - q_{ref,i,p})^2 \quad \text{Eq. 3.6}$$

The distinction between controlled inflows and outflows is made, so a different weight (expressed by the coefficients α) can be given to all objectives independently.

The issue of reaching a desired traffic state is taken into account in the objective function as a penalty rather than a constraint. This is to ensure control freedom for the MPC controller. A constraint can be seen a physical impossibility. In this case, it is unlikely that a subnetwork does not listen to the desired state imposed by the main control. Doing so will be expensive (in terms of the value for the objective function) and therefore very unlikely to happen. However it is not impossible.

3.3.3 Objective formulation: Control to a constant shaped MFD

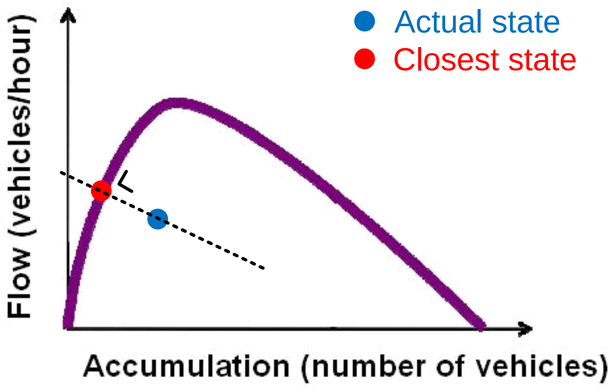


Figure 3.5 – Example of a Macroscopic Fundamental Diagram (MFD). Shortest distance to a point on the MFD

In this formulation, a reference MFD exists that will be used by the upper level controller to communicate the desired traffic state in the subnetwork. A penalty must be applied when a traffic state is calculated that does not exist on the reference MFD. The traffic state consists of values for accumulation (a) and weighted flow (q) (Figure 3.5). Since the shape of the MFD should remain constant, a penalty should be applied both when a value for the traffic state is lower (like the blue dot in Figure 3.5) and higher with respect to the reference values. It must be said that a higher value is unlikely. In that case the subnetwork can either accumulate more vehicles while maintain a certain flow or at the same level of accumulation the flow is higher. In both cases the subnetwork performs better than expected and it should be considered to adjust the reference MFD because it will increase the traffic performance.

An appropriate way to describe the penalty mathematically is by a quadratic function just like the previous case in Section 3.3.2.

$$\min(J_{MFD}) \quad \text{Eq. 3.7}$$

$$J_{MFD} = \alpha_4 \left[(q - q_{ref})^2 + (a - a_{ref})^2 \right] \quad \text{Eq. 3.8}$$

To be able to calculate this objective function, the prediction model should be able to provide flows and accumulation as output. Then, before the value of J can be calculated, the point (a_{ref},q_{ref}) must be determined. The strategy of choosing a reference point that lies closest to the actual state is straightforward. This is the point on the graph where a line perpendicular to the graph intersects with the measured state (see Figure 3.5).

3.3.4 Combined objectives

When all the objectives are combined, the objective function reads:

$$\min(J) \quad \text{Eq. 3.9}$$

$$J = \alpha_1 J_{TTS} + \alpha_2 J_{Qref,out} + \alpha_3 J_{Qref,in} + \alpha_4 J_{MFD} \quad \text{Eq. 3.10}$$

The coefficients α_i will be used to balance the different terms in the final objective function. The values cannot be calculated but should be determined by a trial-and-error process.

In advance, one should determine if one (or more) of the objectives is most important and to which extent. In this thesis, reaching a desired state is more important than optimizing internal performance. As said before, the objective of controlling to a reference MFD will not be tested in this thesis. Therefore the value of α_4 will be 0. The other values will be assigned when explaining the experiments that will be conducted (see Section 3.5).

3.4 Study area

Currently, an interesting issue with respect to traffic control in the Praktijk Proef Amsterdam is the issue of coordination between urban traffic lights and ramp metering. Therefore, the chosen network should contain urban roads and one of the ramps from the A10 motorway in Amsterdam.

Locations where congestion is most likely to occur are more interesting for research when compared to locations where congestion is less likely to occur. At places where congestion is most likely to occur, the traffic could be called "busy". The busiest parts of the A10 motorway are the west and south sections. All ramps connect to busy or less busy urban roads. The S102 ramp at the A10 west connects to an urban road which is one of Amsterdam's four radial corridors into the city center. To the west, the S102 leads to two provincial roads (N200 to Haarlem and N202 to IJmuiden). Therefore, the S102 ramp is chosen as a research location (see Figure 3.6).

The network consists of the S102 ramp and both off-stream intersections. To the west is intersection Basisweg-La Guardiaweg and to the east is intersection Transformatorweg-Kabelweg. In Figure 3.7, a schematic representation of the network is given. The points where traffic can enter or leave the network are indicated with arrows indicating the driving directions. The traffic control can be schematized by four regular traffic light installations and two ramp metering lights. The intersection in the middle (S102 ramp with Basisweg) is one big intersection in reality, but is schematized as two smaller intersections. This is because the prediction model is not capable of handling intersections with this lay-out (see section 4.4). If only TTS is considered as an objective, this simplification should not make a difference in traffic performance. The size of the network is chosen to prevent high computational expenses when performing experiments. The experiments are described in section 3.5.

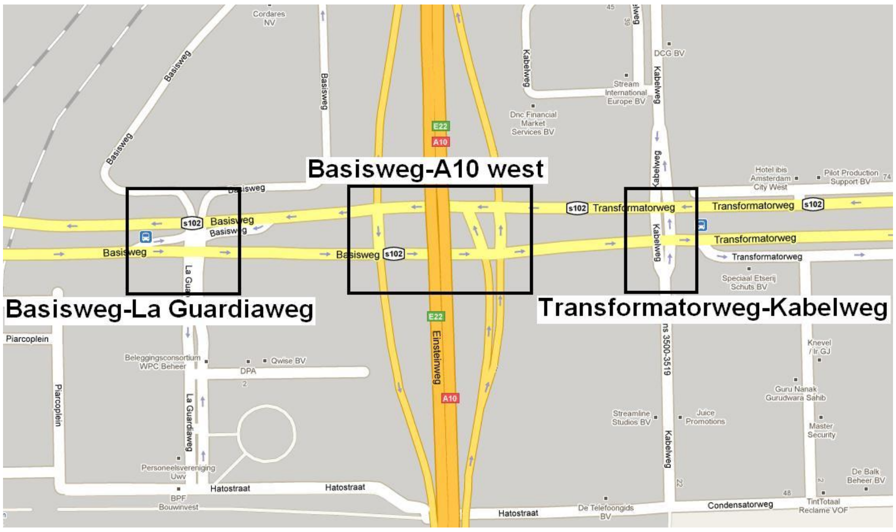
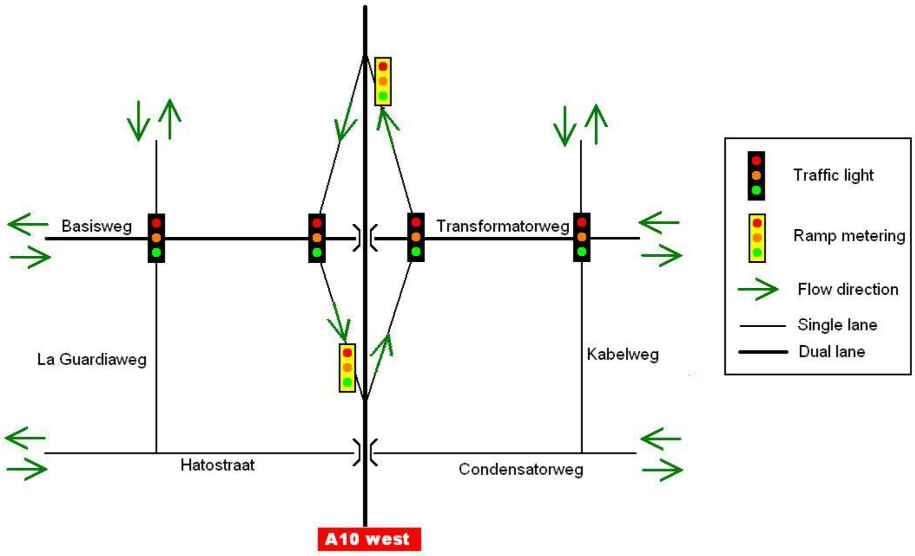


Figure 3.6 - Overview of the study area with three intersections

Figure 3.7 – Schematic representation of the S102 network



3.5 Description of the experiments

Experiments will have to be conducted to find out whether the hierarchical control concept is viable concept, whether MFD is constant or not (and if yes, under which conditions). In this section, the traffic scenarios are described that will be subject to the subnetwork controller. There are many factors that influence traffic dynamics in general. In this thesis, the following six factors are most relevant:

- route choice
- network lay-out
- driver/vehicle behaviour
- traffic light strategy
- traffic load pattern
- desired traffic state

All six factors will be described briefly.

Route choice

Route choice is relevant when road users have multiple routes to choose from. The choice they make has influence on the traffic dynamics of the road. For example, there are two routes available from A to B with similar capacities. From all road users, 67% selects route 1 and 33% selects route 2. Both routes will have different travel times and therefore a different performance. In this thesis, route choice is assumed constant by offering just one route alternative per origin-destination pair. The reason not to vary this factor is because of limitations in the prediction model; the link Hatostraat-Condensatorweg (see Figure 3.7) cannot be modelled.

Network lay-out

A grid-like road network performs differently when compared to a radial network. Whether roads are one- or bi-directional has an influence on traffic performance. The number of lanes per direction, the amount of through and turning lanes in front of a (controlled) intersection, all have influence on the traffic dynamics. Because of the available time, one network will be considered in this thesis. The network used is described in section 0.

The choice for this network will certainly have consequences for the shape of the MFD, since each road network had its own MFD.

Driver/vehicle behaviour

If all other factors remain constant, it is possible to find different results in terms of traffic performance for different traffic models. Examples are Fosim, Wiedemann Model and Intelligent Driver Model. In this thesis, one model will be selected because the time available is not sufficient to test more. Also, testing more models will very likely not have an additional contribution with respect to the research questions.

Traffic light strategy

A different strategy means different performance in traffic. Vehicle actuated control measures the traffic real-time and is therefore capable of quickly responding to the actual traffic situation. This way of controlling is good in minimizing the average waiting time (before the signal turns green). Fixed time traffic control does not measure actual traffic presence and can therefore not respond to the actual traffic situation. On the other hand, it is relatively easy to generate traffic schemes that allow coordination between different traffic lights. This way of controlling can be advantageous for locations where the same traffic patterns occur repeatedly and where coordination is beneficial. A combination of vehicle actuated control and coordination requires a controller more advanced compared to the vehicle actuated and fixed time controller, for example an MPC controller. All three principles are present in this thesis, see Section 3.5.2.

Traffic load pattern

Different traffic load patterns correspond to periods of different demand. For example:

- Peak hours
- Off-peak hours
- Event
- Incident

These different demand patterns have different characteristics. From a control point of view, it can be expected that different traffic load patterns require different control strategies to perform optimal. Off – peak hours are not expected to be advantageous with respect to MPC. Incident traffic cannot be modelled by both the prediction model and the microscopic model. So the traffic load patterns used in this thesis are peak hour and event traffic. They are described in more detail in section 3.5.1.

Desired traffic state

The higher level control can ask a subnetwork to reach a desired state. Because of this, the subnetwork will perform different when compared to a situation where it does not have to take into account desired states from an upper level control. Different desired traffic states will be applied in this thesis. More on this can be found in section 3.5.3.

In sum, route choice, network lay-out and driver behaviour will be considered constant. Varying these would cost too much extra time. For the driver behaviour and the network lay-out, the reasons are straightforward. The choice for constant route choice has been made because otherwise a variation in route choice can be responsible for a different traffic performance. This will be hard to measure so it is better to eliminate that possibility. So, the traffic load pattern, the traffic light strategy and the desired traffic state will be varied in two experiments. More information on the desired traffic state can be found in section 3.5.3. The content of both experiments will be explained in sections 3.5.2 and 3.5.3. Both experiments will be subject to two different traffic load patterns, which will be described in section 3.5.1.

3.5.1 Traffic load pattern

In section 0, the interesting nature of “busy” traffic was indicated. The following two traffic load patterns are likely to cause congestion (can be considered busy) and are likely to occur in reality:

- Regular peak hour traffic
- Event traffic (for example a concert at Westergasfabriek)

Regular peak hour traffic happens roughly 220 days per year (44 weeks with 5 working days, remaining 8 weeks in the year are holidays with less busy traffic). Therefore, this situation is worth looking at. The traffic loads are typically not the same on all working days. In this experiment, one peak hour pattern will be assumed because of the available time to conduct experiments. Event traffic happens on a regular basis, although not as often as peak hour traffic. One can think for example of concerts and football games. At this particular location in Amsterdam, event traffic because of football games is highly unlikely. But the results give an indication of which performance can be expected when using an MPC controller at locations where this phenomenon does happen.

Both load patterns are hypothetical and they will be applied for duration of two and half hours. This is because a typical peak period has duration of two hours. In a microscopic model, the network needs to be filled with traffic, 30 minutes is sufficient to do so. The peak hour flows can be seen in Figure 3.8. The event traffic flows can be seen in Figure. All flows are in vehicles per hour. For both traffic load patterns, the flows can be scaled by multiplying all flows with the same factor. In Figure 3.10 and Figure 3.11, the scaling factors for the peak hour and event loads can be found. The loads are chosen with an early peak, so congestion will occur. Then the network can recover from the congestion because the traffic load is lower again. The occurring of congestion is necessary, because with absence of congestion one can only find the free flow part of the MFD, which is not so interesting.

Figure 3.8 - Traffic flows in peak hour loading pattern

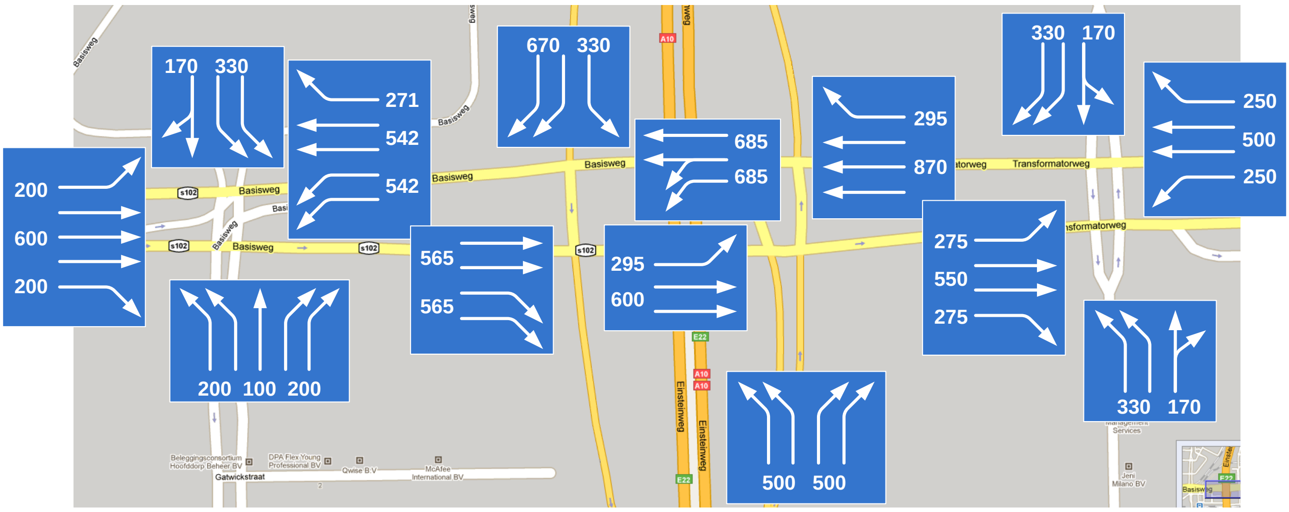


Figure 3.9 - Traffic flows in event loading pattern

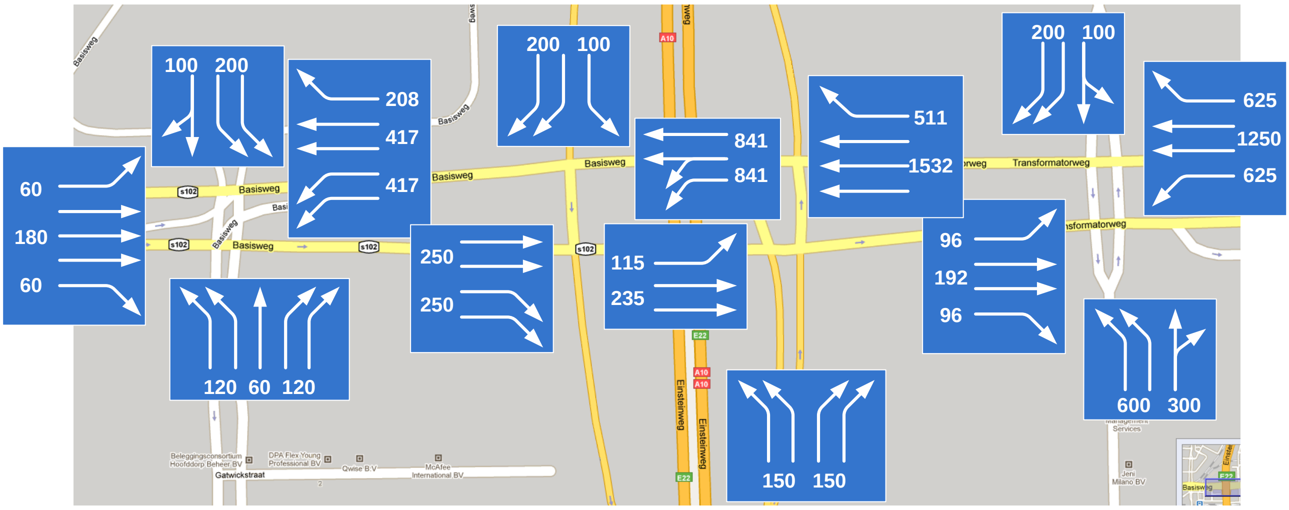


Figure 3.10 – Scaling of the flows in peak hour traffic

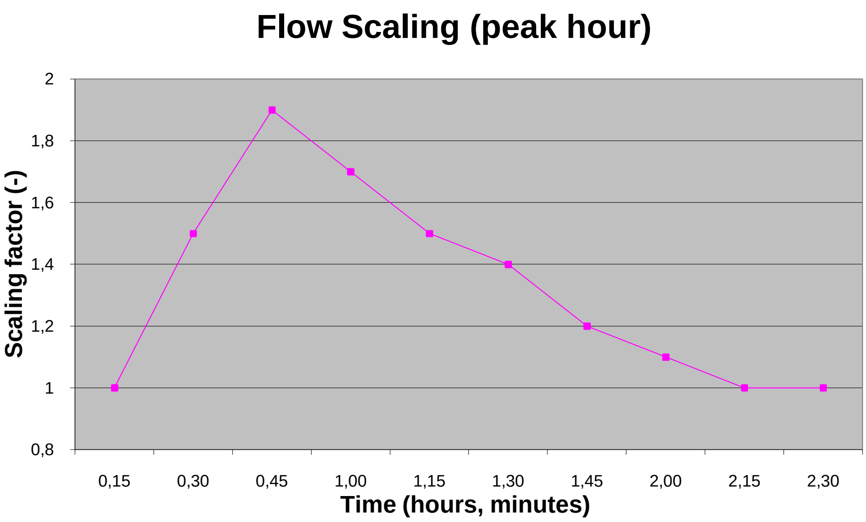
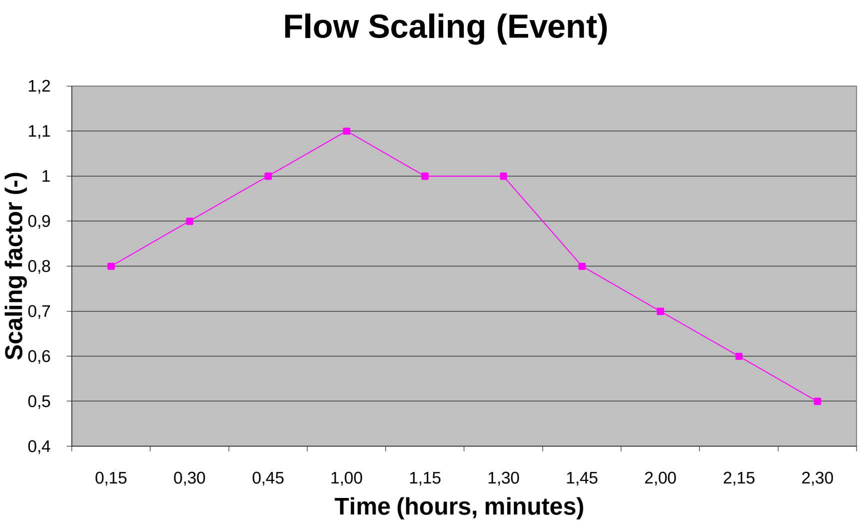


Figure 3.11 – Scaling of the flows in event traffic



3.5.2 Experiment 1

In this experiment, both traffic patterns described in section 3.5.1 will be subject to the following control strategies:

- Fixed time signal control
- Vrigen/Trafcod control method
- MPC signal control

Recall research questions 2 and 3:

Main research questions

2. What is the difference in performance between coordinated control and conventional control methods?
3. Is it possible to use MFD in a control concept for subnetworks?

The second question will be answered with this experiment, and it is likely to provide a partial answer for the third question. The contribution to the third question is the fact that MFD's will be generated, to investigate the dynamics of the MFD in the S102 network. This means, is the MFD constant and if yes under what conditions? The traffic performance is indicated by the total time spent (TTS). The TTS for all three control strategies will be compared.

No ramp metering is applied is this experiment (this will be considered in the next experiment), in this experiment traffic signal control is considered without higher level objectives. Ramp metering could be seen as a higher level objective, it removes control freedom.

3.5.3 Experiment 2

In this experiment, only MPC signal control will be considered. The traffic load patterns are described in section 3.5.1. In this experiment, the desired state will be varied. The ultimate goal for Praktijk Proef Amsterdam would be a constant MFD for a subnetwork. The scope of this experiment is to answer whether the shape of the MFD is constant or not when desired traffic state(s) are applied. Recall the higher level control (Section 3.2) that wants to communicate a desired state to a subnetwork by means of a dot on the MFD. Possible actions derived from a desired state in a subnetwork could be:

1. request a desired (maximum) outflow to an adjacent network
2. request a desired (minimum) inflow from an adjacent network

For urban networks, these two actions correspond to signalled intersection control on the edge of a subnetwork. When translated to highways, requesting a maximum outflow corresponds to the working of on-ramp metering. Requesting a minimal inflow could be done to prevent traffic jams because of long queues on the off-ramp. The values for the desired inflows and outflows are assumed constant in this experiment; they do not change in time. In theory, this does not have to be the case. It is important to realize that the traffic inside the subnetwork will be optimized. The settings of for example ramp metering are not optimized but considered to be given.

In this experiment, only the situation is considered in which the S102 subnetwork gets desired states which are not in favour of the subnetwork. It is not considered that the S102 subnetwork can make a request to limit outflow from another subnetwork into the S102 subnetwork.

For the S102 subnetwork in Amsterdam, two directions are chosen to be limited (see Figure 3.12). Technically it is possible to control any direction (both green and red arrows in Figure 3.12). The reason to choose a limited number of directions has to do with the time budget; it would take too much time to investigate all possibilities.

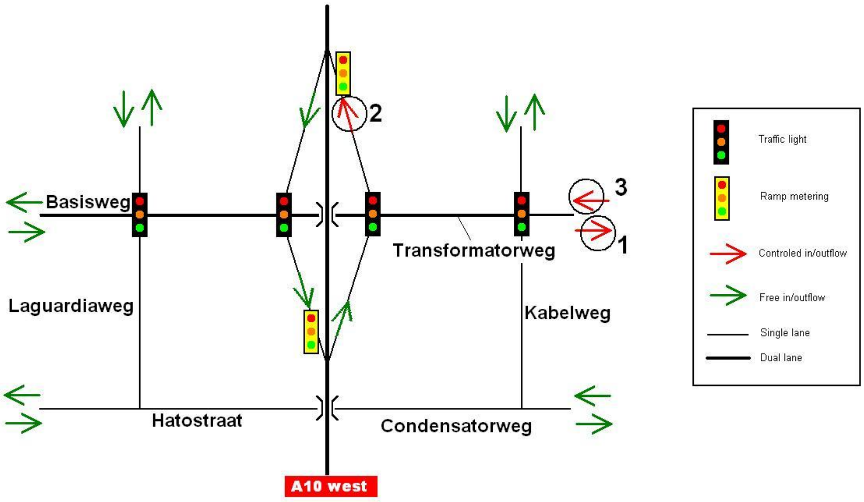


Figure 3.12 – Boundaries in the network, with or without controlled flows.

The different scenarios that will be conducted are:

1. Limited outflow at Transformatorweg (peak hour)
2. Limited outflow to A10 west in northern direction (peak hour)
3. Minimal inflow from Transformatorweg (event)

Note that ramp metering will be applied in scenarios 1 and 2 (the ramp metering to the A10 in southern direction as seen in Figure 3.12 is not used).

The performance indicators are:

- total time spent
- the difference in realized and reference flow at the borders where a desired flow was specified

Also, MFD's will be generated to investigate the dynamics of the MFD in this urban network (is it constant, if yes, under which conditions). This experiment answers another part of research question 3. When compared to experiment 1, there is the influence of higher level objectives. They could have influence on the shape of the MFD.

Since both experiments are described now, the coefficients for the objective function can be given. The objective function is repeated and the coefficients are listed in Table 3.1.

$$\min(J) \qquad \text{Eq. 3.9}$$

$$J = \alpha_1 J_{TTS} + \alpha_2 J_{Qref,out} + \alpha_3 J_{Qref,in} + \alpha_4 J_{MFD} \qquad \text{Eq. 3.10}$$

Table 3.1 – Values of coefficients for the objective function

coefficient	Experiment 1	Experiment 2
α_1	1	1
α_2	0	0.001
α_3	0	0.001
α_4	0	0

In experiment 1 (see section 3.5.2), only the first term (TTS) is present. In experiment 2 (see section 3.5.3) three terms are present. The values for experiment 2 are chosen in this way, because the absolute value of $J_{Qref,out}$ and $J_{Qref,in}$ can reach such high values that the value of J_{TTS} becomes negligible. This was found by trail-and-error. The fourth term is in both cases zero, because time constraints did not allow performing an experiment where the MPC controller is penalized for deviating from a reference MFD.

4. Simulation Environment

4.1 Introduction

In this chapter, the choices for all parts of the simulation environment will be explained. Also, their specific characteristics (like parameters) will be discussed. This is part of the research approach. However, because of the size of this subject, the choice was made to give it its own chapter.

The terms cycle time and traffic scheme will be important in this chapter. A general explanation on these items is given in Section 2.5. The cycle time that will be used for all cases is 90s. Sometimes this is the maximum cycle time. If so, this is clearly indicated. The duration of 90s is chosen because the network lay-out of the S102 network (see Section 3.4) is such, that the general maximum cycle time (in The Netherlands) of 120s is not efficient. The intersections in the study area are in close proximity (distances are around 100 meter with respect to each other), so the buffer space for waiting cars in front of a red traffic signal is limited. In case of a cycle time of 120s, the likelihood is higher that an intersection will be blocked. This decreases the network performance.

In Section 4.2, an overview of the simulation environment is given.

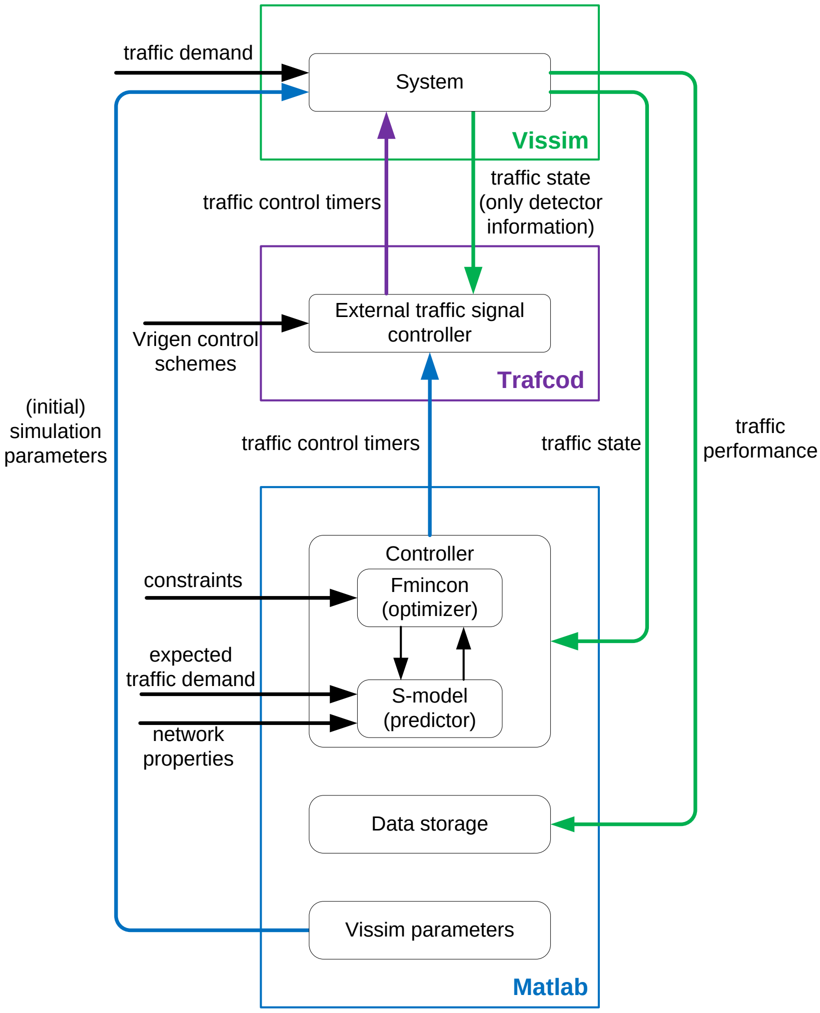
4.2 Overview of the simulation environment

To conduct experiments (as described in Section 3.5), the following software has to cooperate simultaneously:

1. Matlab
2. S-model
3. Vissim
4. Trafcod

The interaction of this software is given in Figure 4.1. One typical cycle will be described step by step. One cycle in the diagram is equivalent to one cycle time of the traffic signal control during simulation.

Figure 4.1 – Structure of the simulation environment



1. Vissim will simulate for one cycle time and pause the simulation.
2. The traffic state and the traffic performance are measured. The traffic performance is stored in Matlab. The traffic state is used as input for the controller.
3. The S-model (prediction model) is started, with the general network properties (link length, number of lanes, etc.) and expected traffic demand. The dynamic network properties are updated with the traffic state information from Vissim.
4. The optimizer will optimize the control signal. The constraints involved are cycle time, minimum and maximum green times.
5. The optimized result cannot be directly implemented in Vissim. An external traffic light controller is required to do so. From the controller in Matlab, files are created with the new timer information. Trafcod checks 10 times per second if new files are available to absorb the new timer values. When found, the timers are adjusted and the files are deleted. Trafcod uses traffic control schemes generated in advance by the program Vrigen.

6. Matlab will tell Vissim to run again for one cycle. During the cycle time, Trafcod controls the traffic signals present in Vissim.
7. Repeat this cycle, go back to step 2.

4.3 Programming software (Matlab)

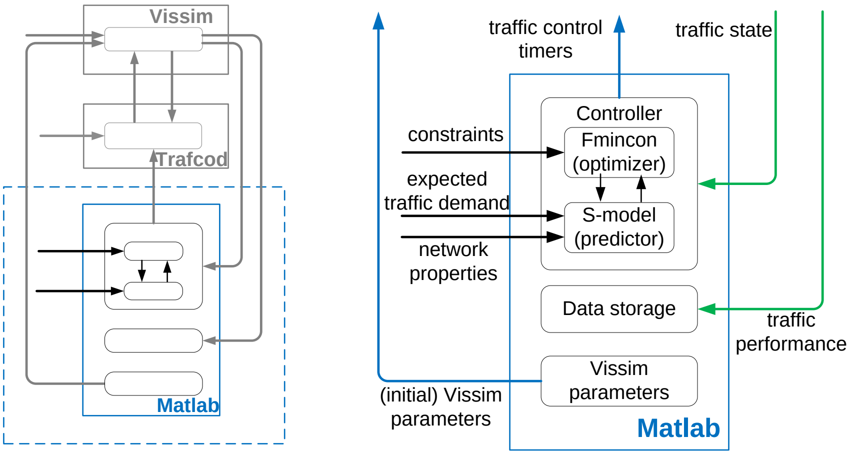


Figure 4.2 – Position of Matlab in the simulation environment

In this section, the choice for Matlab as the heart of the control software will be explained and the structure of the two main files that control the simulation environment will be given. It can be considered the heart of the software, since all other software can be controlled in a central way by Matlab.

4.3.1 Different options

Using a programming language like C, C++ or Java is unnecessary complicated when compared to FORTRAN and Matlab. The latter two are specifically designed for technical computing. Although the author has prior experience with Java, too much time would be spent on learning the programming language and writing functions that are already available in FORTRAN and Matlab.

When comparing FORTRAN and Matlab, they look alike. Some functions (like plotting graphs) are not native in FORTRAN. Both packages are portable; they work on different platforms (Mac OS, Linux and Windows). Since the author has prior experience with Matlab and software licenses are available, Matlab is chosen as the programming software.

4.3.2 Working of the controller

The Matlab code is written in such a way that most calculations and actions are written in functions. A good example is the detection of vehicles on all links after one cycle is completed. This code extracts all vehicle data and changes the data format, so the prediction model can read it. The code consists of about 600 lines. In the code one level higher, this is then reduced to the following command:

```
[output] = linkUpdate(input)
```

In this way, the structure of the code is easy to read and understand. The two upper level files will be described here.

The upper level file (run_MPC_batch.m) contains initial parameters and gives the possibility of running a batch of runs. The initial parameters given here are for example the traffic load scales (see Figure 3.10), random seed and some MPC parameters. The only parameter that will be changed for each run is the random seed. For more information on random seed, see section 4.8.4. After all runs are completed, data is stored and graphs are plotted. Schematically, this code looks as follows:

```
[initial Vissim parameters]
[initial MPC parameters]

for (all random seeds):
      run_MPC(random seed, initial parameters)
end

[store data]
[plot graphs]
```

There is only one function is this upper level code, namely the function run_MPC. This is the second level of the code. The structure of run_MPC is as follows:

```
[other fixed initial parameters]
[initialize Vissim]

for (all cycle times of the simulation)
      [run Vissim for one cycle]
      [extract and process data to plot MFD]
      [extract travel times]
      [measure traffic state in Vissim and pass it on
      to S-model]
      [initialize S-model]
      [run S-model]
      [delete S-model]
      [adjust initial solution (x_0) for S-model]
      [adjust timers for Trafcod]
end
```

The working of the S-model will be explained in section 4.4.

4.4 Prediction model (S-Model)

A general description of the model can be found in Appendix A. The S-model was chosen as the prediction model for MPC because it is capable of calculating the TTS of in urban traffic network relatively fast, much faster when compared to a micro simulation model. The S=model needs less than one second to compute TTS where a micro simulator would need roughly three and a half minutes (210s). Since traffic is a dynamic process, it is desirable that the most recent traffic state information can be used. In order to do so, computational time to calculate an optimal control signal should be a short as possible.

In this section, first the control horizon and the prediction horizon will be discussed. Then, the modelling of the network (4.4.2) and the control phases (4.4.3) will be described.

4.4.1 Control horizon and prediction horizon

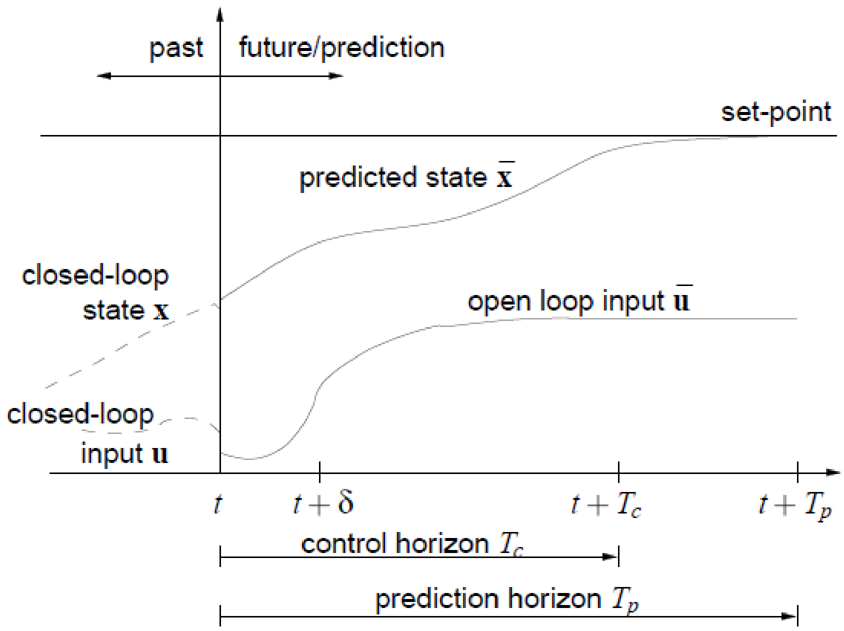


Figure 4.3 – Control horizon and prediction horizon (Findeisen and Allgöwer, 2002)

A general description of control and prediction horizon can be found in section 2.2. The prediction horizon should be chosen such, that all traffic dynamics can be predicted. This means the horizon should be long enough for a vehicle to both enter and leave the network. The size of the network is relatively small; the longest distance that could be travelled in the S102 network is about 3.5km. The maximum speed is 50km/h, so without traffic lights this means travelling for 4:12 minutes. There are four controlled intersections to pass. Suppose everywhere, a vehicle has to wait for two complete cycles to pass the intersection. Then the trip inside the network took 16:24 minutes. To be able to follow this vehicle with a relative long travel time, the prediction horizon should be set longer than 16.24 minutes. The prediction horizon is therefore set at 22.30 minutes. Both the prediction and the control horizon in the S-model are multiples of the cycle time. The prediction horizon N_P is thus 15 cycle times (22.30 minutes). The control horizon can be smaller with respect to the prediction horizon to save computational time and to force the controller to take action early in the process. The control horizon T_C is set at five cycle times.

4.4.2 Network modelling

Figure 4.4 – Schematic representation of the S102 network

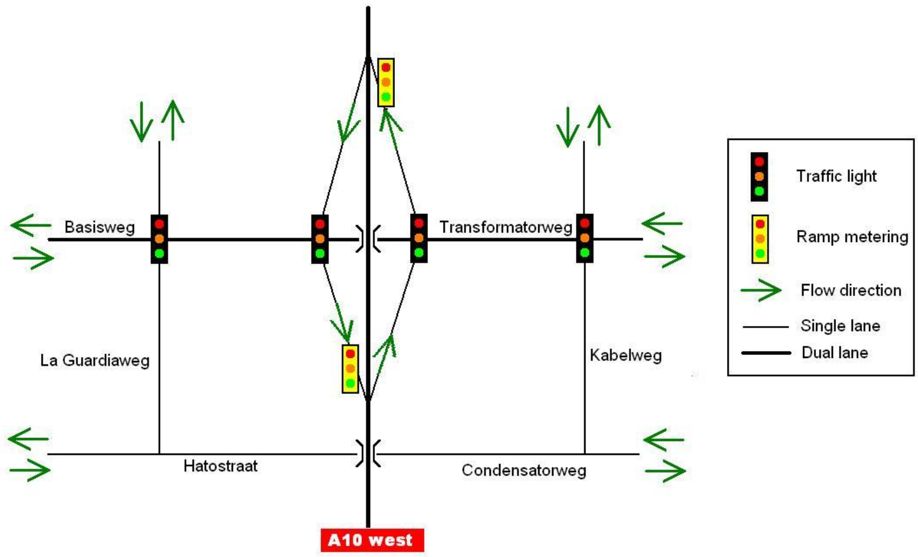
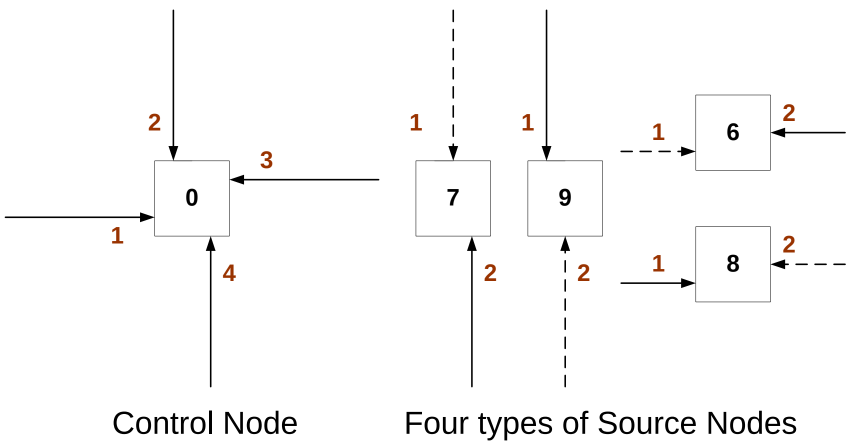


Figure 4.5 – Node types (black) and link numbering convention (brown)



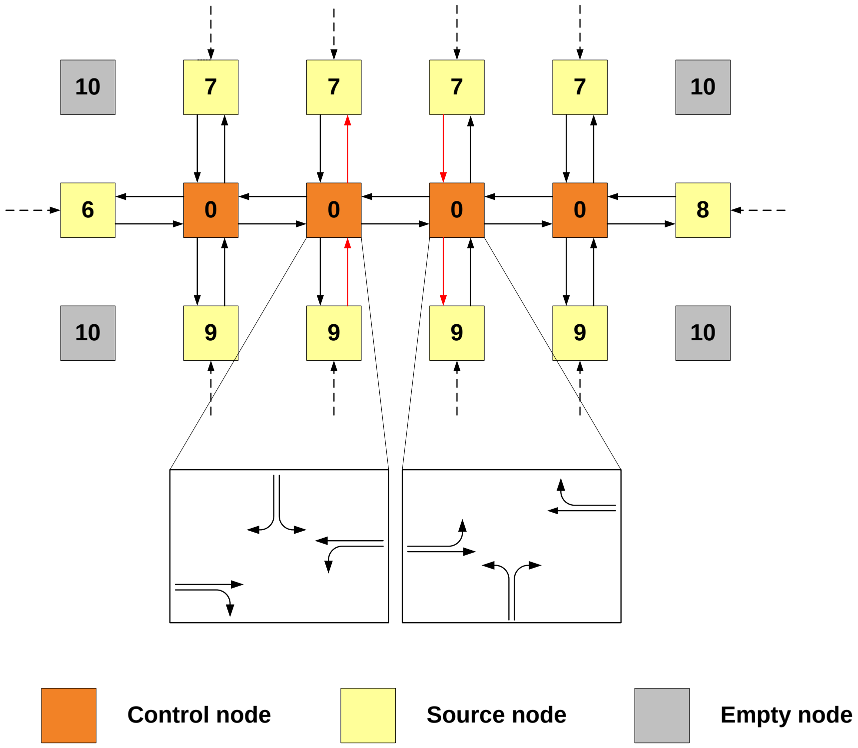


Figure 4.6 – Schematization of the S102 network for the S-model. Dashed and red links do not exist in reality

The S-model was designed to calculate traffic state and performance for a grid network. The S102 network (Figure 4.4) however is not a grid network. This section explains the way the S102 network is modelled.

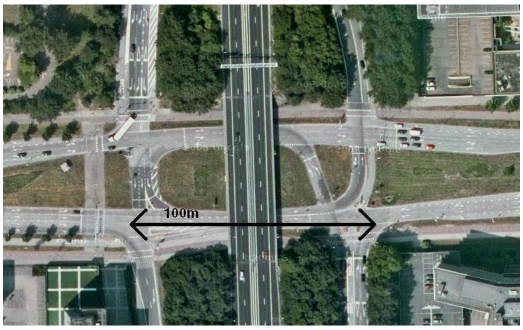
The basic elements that are use to built a network are given in Figure 4.5. These elements are links and nodes. Control nodes are intersections (indicated with node type 0) and source nodes are at the edge of the network (indicated with node type 6 until 9). The shape of the network is communicated to the S-model as a matrix. With the building blocks described, voids could appear. They are filled with nodes of node type 10 (which have no physical meaning). The other numbers are used for node types not used in this thesis (for example t-junctions).

Between the nodes there are links. Every node (as a building block as seen in Figure 4.5) has only incoming links. Those links are the outgoing links for the adjacent nodes. A control node has four incoming links. A source node has two incoming links. The link indicated with a dotted arrow is a virtual link, there is no physical meaning. With these "building blocks" an infinite large grid network can be created.

The convention for counting the nodes is from left to right for each row, starting at the highest row. For the links, the numbering convention per node can be seen in Figure 4.5. The complete numbering of the S102 network for the S-model can be found in Appendix C.

In Figure 4.6, the schematization of the S102 network for the S-model can be seen. The big intersection with the on- and off-ramps is modelled as two smaller separate intersections. Considering one intersection would not be appropriate since the model takes into account fixed clearance times of 2 seconds for all conflicts. Since the distance between the two parts is almost 100 meter (see Figure 4.7), the clearance times are very high when considering only one intersection (around 13 seconds for the left turning movements).

Figure 4.7 – Distance in the intersection Basisweg – A10 ramp



In total, there are 36 links in the network (see Figure 4.6 and Appendix D). Every link has 26 parameters which can be adjusted. The complete overview can be found in the code (the file linkdata.m), as well as the desired format. The most important characteristics will be discussed here. Those are:

- Capacity (the maximum amount of vehicles that can be inside the link)
- Free space (the capacity minus the amount of vehicles present in the link)
- Amount of waiting vehicles in front of a traffic signal (separated into traffic waiting to turn left, right and to go through)
- The turning rates (a% through going, b% left turning, c% right turning)
- Saturation flow (maximum flow when a green signal is given)
- The phase sequence for the traffic signals

The turning rates and saturation flows are coupled in the S-model. This means that to assign proper saturations flows, the following should be done. It is assumed that one lane has a saturation flow of 1800 vehicles per hour. A decrease with respect to the saturation flow because of the turning radius is ignored. In Figure 4.8, both links have a saturation flow of 7200 vehicles per hour, because they both have four lanes. To give the right saturation flow rate to each direction, the turning rates should be as indicated. This has important consequences for the traffic load pattern, as this should match the turning rates derived from the correct assignment of saturation flows. The traffic load patterns (see section 3.5.1) are thus chosen in such a way that the turning rates match the road geometry.

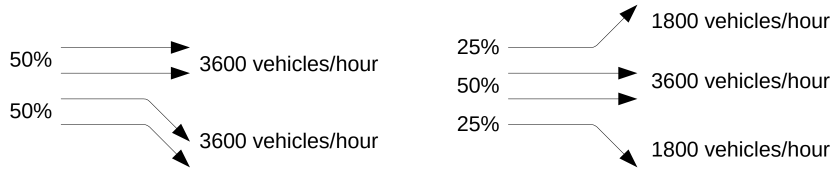


Figure 4.8 – Relation between saturation flows and turning rates. Two examples

The traffic load pattern can be calculated when the inflow at each source is known. This can be done relatively easy with the help of Excel (see the file flows_turnrates.xls).

4.4.3 Phase modelling

The phases order is determined with help of Vrigen. The phase sequence will be shown for each intersection separately.

Basisweg-La Guardiaweg

Table 4.1 – Phase sequence for Basisweg-La Guardiaweg

Phase	Signal groups green
1	1, 6, 7, 12
2	7, 8, 9
3	1, 2, 3, 4
4	4, 5, 11

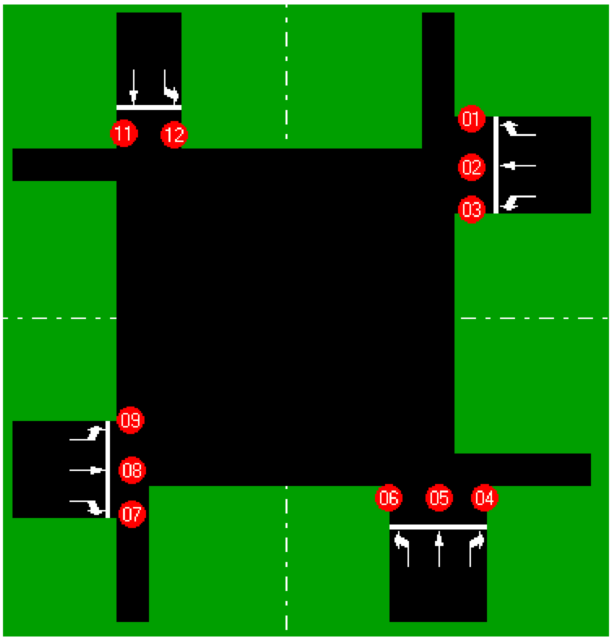


Figure 4.9 – Signal groups at intersection Basisweg-La Guardiaweg

Basisweg-A10 (southbound)

Table 4.2 – Phase sequence for Basisweg-A10 (southbound)

Phase	Signal groups green
1	2, 63
2	7, 8, 10
3	7, 10, 12, 72

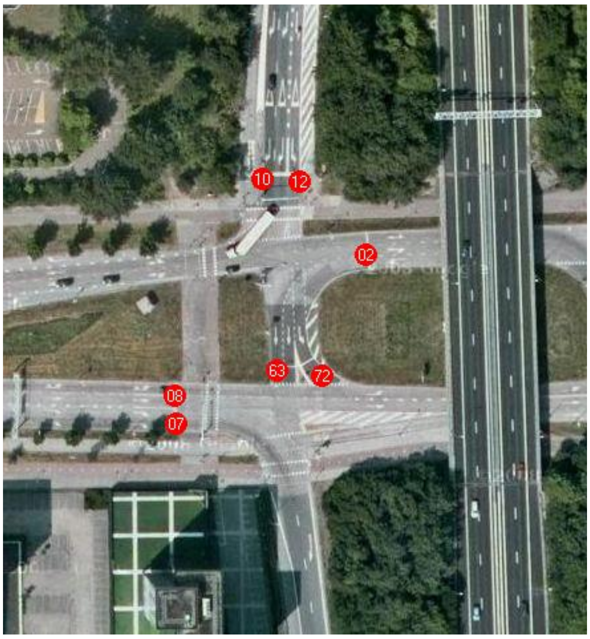


Figure 4.10 – Signal groups at intersection Basisweg-A10 (southbound)

Table 4.3 – Phase sequence for Basisweg-A10 (northbound)

Basisweg-A10 (northbound)

Phase	Signal groups green
1	1, 2, 8
2	1, 4, 6
3	8, 9

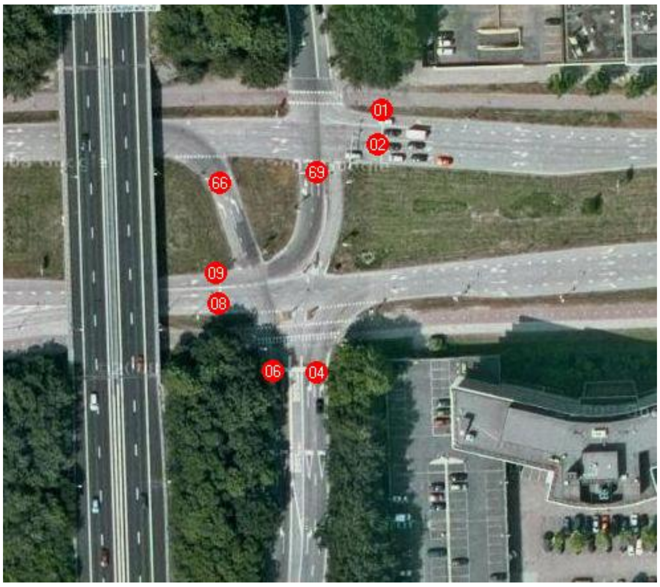


Figure 4.11 – Signal groups at intersection Basisweg-A10 (northbound)

Table 4.4 – Phase sequence for Kabelweg-Transformatorweg

Kabelweg-Transformatorweg

Phase	Signal groups green
1	1, 10, 11
2	7, 8, 9, 10
3	1, 2, 3
4	5, 6, 7

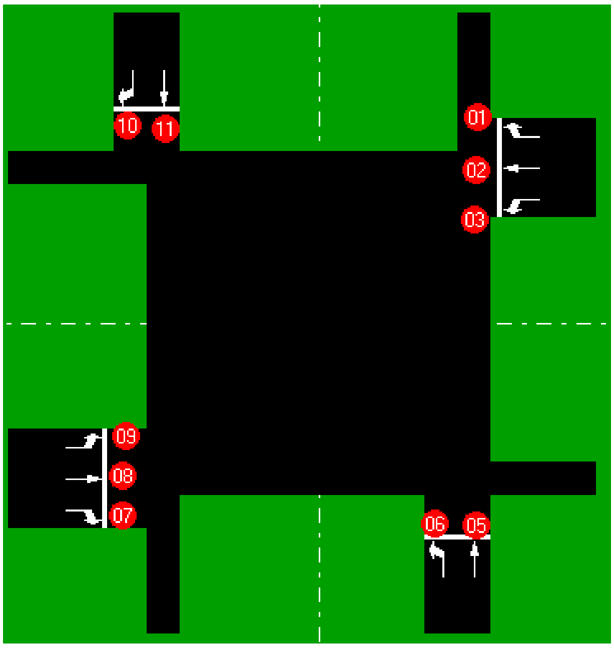


Figure 4.12 – Signal groups at intersection Kabelweg-Transformatorweg

4.4.4 Control variables

The S-model calculates the traffic in a network, given the green times for each phase. Each green time calculated includes 3 seconds amber time and 2 seconds clearance time. Based on the network lay-out, traffic flowing into the network and the green timers, the S-model will calculate the total time spent in the network as a result. Traffic flowing out of the network is normally considered unlimited. In the second experiment, the outflow is limited due to ramp metering or limited flow on an urban road. In that case, the S-model has also limitations incorporated.

The S-model considers the cycle time to be constant. This is a serious limitation in control freedom, since some scenarios cannot be improved in terms of traffic performance. It is possible that improvement in the traffic performance can only be reached by a lower cycle time if the traffic demand is significantly lower than the capacity of an intersection. On the other hand in traffic conditions where congestion is likely, a (temporary) increase of the cycle time may improve the traffic performance.

The amount of control variables depend on the number of intersections (control nodes) and the number of phases per intersection. For the S102 network, the intersections Basisweg-Laguardiaweg and Kabelweg-Transformatorweg (see Figure 4.4) have 4 phases each. The two intersections in the middle which account for the Basisweg-A10 ramp intersection (see Figure 4.7) have three phases each.

The S-model calculates the green time for all but the last phase of each intersection. This last phase can be calculated, because the cycle time is fixed (at 90s). So, in total there are 10 variables (see Table 4.5) to be controller per cycle time. The total number of control variables depends on the control horizon. The control horizon is 5 cycle times, so the amount of control variables in one prediction is 50.

Table 4.5 – Number of phases and control variables per intersection

Intersection	#phases	#control variables
Basisweg-Laguardiaweg	4	3
Basisweg-A10 (southbound)	3	2
Basisweg-A10 (northbound)	3	2
Kabelweg-Transformatorweg	4	3

4.4.5 Objective function

In section 3.3, the formulation of the objective function is explained. For the sake of easy reading of this document, the MPC formulation is repeated briefly in this section. Recall the general formulation of the objective function in Eq. 3.9 en Eq. 3.10:

$$\min(J) \quad \text{Eq. 3.9}$$

$$J = \alpha_1 J_{TTS} + \alpha_2 J_{Qref,out} + \alpha_3 J_{Qref,in} + \alpha_4 J_{MFD} \quad \text{Eq. 3.10}$$

Table 4.6 – Values of coefficients for the objective function

coefficient	Experiment 1	Experiment 2
α_1	1	1
α_2	0	0.001
α_3	0	0.001
α_4	0	0

In experiment 1 (see section 3.5.2), only the first term (TTS) is present. In experiment 2 (see section 3.5.3) three terms are present. The values for experiment 2 are chosen in this way, because the absolute value of $J_{Qref,out}$ and $J_{Qref,in}$ can reach such high values that the value of J_{TTS} becomes negligible. This was found by trail-and-error. The fourth term is in both cases zero, because time constraints did not allow performing an experiment where the MPC controller is penalized for deviating from a reference MFD.

4.4.6 Optimization of the objective function

The optimization of the objective function is done with the function "fmincon" from the Matlab optimization toolbox. The inputs are a custom function or model, constraints, upper and lower bounds and an initial solution. All items will be explained.

Custom model

This can range from a simple equation like $y = b(x-a)^2$ to a complex model that describes aircraft dynamics. The model used here obviously is the S-model. Fmincon will find a (local) minimum for the objective function by calculating the optimal values for all control variables.

Upper and lower bounds

Every control variable can be given a minimum and a maximum value. The control variables are the green times. Table 4.7 gives the values for all four control nodes. The maximum green time is derived from the cycle time minus three times the minimum green time. The cycle time is chosen at a fixed value of 90 s, see the introduction of this chapter. A common value for the minimum green time in The Netherlands is 6 seconds. The S-model includes amber light (3 seconds) and clearance time (2 seconds), so the minimum green time for the S-model is 11 seconds. The higher minimum green time for the Basisweg A10 intersections are chosen because the traffic streams are busier at these intersections. Both the upper and lower bound values are offered to fmincon as a 50 by 1 vector.

Table 4.7 – Minimum and maximum green times. All timers are in seconds

Intersection	Minimum green time (s)	Maximum green time (s)	Cycle time (s)
Basisweg-Laguardiaweg	11	57	90
Basisweg-A10 (southbound)	11	68	90
Basisweg-A10 (northbound)	11	68	90
Kabelweg-Transformatorweg	11	57	90

Constraints

To save computational time as described in section 4.4.4, constraints should be formulated. Fmincon accepts constraints in the following format, both equalities and inequalities:

$$A \cdot x \leq b$$

$$Aeq \cdot x = beq$$

The constraints are derived from the minimum and maximum green times, and the cycle time. The following inequality should be satisfied for the 4-phase control nodes:

$$x_{\min} \leq T_{cycle} - x_{phase1} - x_{phase2} - x_{phase3} \leq x_{\max} \quad \text{Eq. 4.1}$$

This equation should be written such that the format is similar to the format fmincon can read. Eq. 4.1 is rewritten as two inequalities:

$$x_{phase1} + x_{phase2} + x_{phase3} \leq T_{cycle} - x_{\min} \quad \text{Eq. 4.2}$$

$$-x_{phase1} - x_{phase2} - x_{phase3} \leq x_{\max} - T_{cycle} \quad \text{Eq. 4.3}$$

For the 3-phase control nodes, these equations become simpler since they only contain two control variables instead of three. In matrix format, Eq. 4.2 and Eq. 4.3 yield:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & -1 \end{bmatrix}, x = \begin{bmatrix} x_{phase1} \\ x_{phase2} \\ x_{phase3} \end{bmatrix}, b = \begin{bmatrix} 79 \\ -33 \end{bmatrix}$$

$$A \cdot x \leq b$$

Note that the numbers in vector b are different for the 3-phase control nodes.

The sequence of the control variables in the S-model is as follows. Still, a horizon of 5 cycle times is assumed. The first 15 variables describe control node 1 (Basisweg-Laguardiaweg). The next ten variables describe control node 2 (Basisweg-A10 ramp [southbound]). Then, ten variables describe control node 3 (Basisweg-A10 ramp [northbound]). The final 15 variables describe control node 4 (Kabelweg-Transformatorweg). A graphical representation can be found in Figure 4.13, the upper part ("Optimal solution at cycle time x").

Initial solution

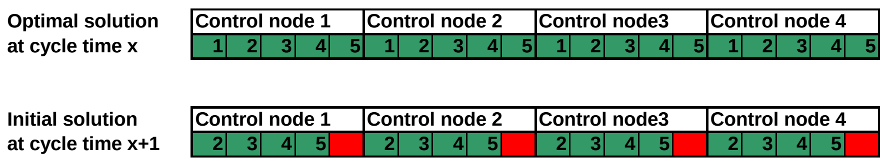


Figure 4.13 – Adjusting the initial solution for the next optimization

Fmincon tries to find a minimum value for an objective function. It does so by iterations. Every iteration, different values are guessed for all control variables, based on their derivatives.

When a function is considered with two independent variables, it is possible to plot a graph and see where the lowest value of the function within the plotted domain is located. Also, it is likely that local minima will be found. It depends on the starting solution whether fmincon will find the absolute minimum or a local minimum. In the S-model, there are 50 independent variables. The chance of reaching an absolute minimum will be small. To increase the chance of finding “the best” local minimum, the following strategy is used.

For every prediction, three initial solutions will be used. The solution with the lowest objective value of all three is considered the optimal solution for this time step. The four different initial solutions are:

- The previously optimal solution, adjust for one time step. This principle is illustrated in Figure 4.13. The last time step is then empty (indicated with red). There are several options how to fill in this time step. One is to calculate an average of the previous timers, another is to copy the timers from the previous time step (just the box with number 5 from Figure 4.13 to the red box). In this thesis, it is chose to fill this gap with reference timers (see the next bullet).
- A reference solution. This solution contains green times from which a good performance can be expected. The times are derived from the fixed time strategy. This initial solution is always the same.
- A solution consisting of a base value plus a random component. The base value is the minimum green time and the random component ranges between 0 and (x_{max}-x_{min})

The computational time for one prediction with a control horizon of 5 (cycle times) and a prediction horizon of 15 (cycle times) depends highly on the flows present in the network. When the flows are low, one prediction can be as short as 20s whereas a prediction for congested traffic may take 180s before the convergence criterion is reached.

4.5 Vrigen control method

The program Vrigen is designed at Delft University of Technology. This program allows building traffic control schemes according to Dutch standards. The schemes can then be executed by the program Trafcod or a CCol based program.
The schemes are generated by the following input:

- Traffic flows
- Conflict matrix with clearance times
- Saturation flow while green
- Timing constraints like maximum cycle time and minimum green times
- Possible relationships between connected green phases

One main objective in Vrigen is to minimize lost internal time. This is done when generating the control schemes (choose most efficient phase order). Another main objective of Vrigen is to minimize waiting time in front of a traffic light, when executing the traffic scheme by Trafcod. In practice this means for example: do not give green when no traffic is present, extend the green phase when traffic is still present. The cycle time set in Vrigen is a maximum cycle time. The value is 90s. When less traffic is present than expected, or if demand has a short drop, the cycle time realized will be shorter. This can be an important advantage with respect to the MPC and fixed time controller which both have fixed cycle times.

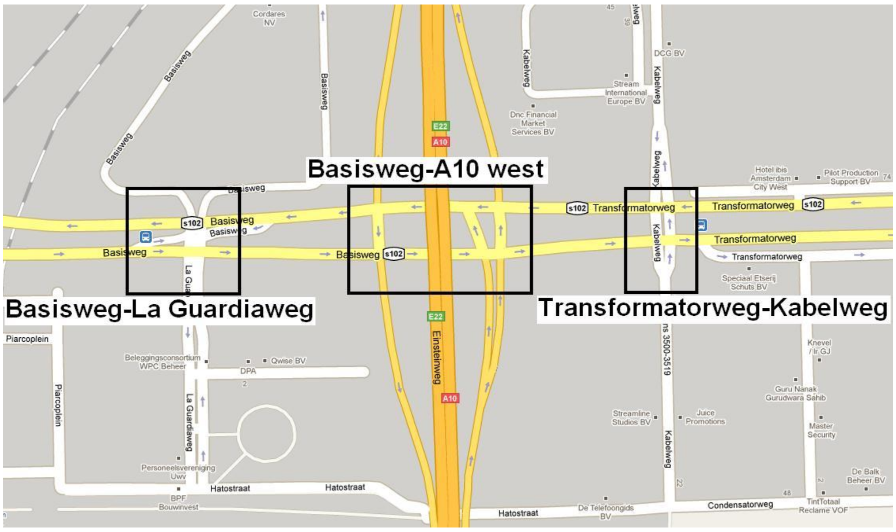


Figure 4.14 - Overview of the study area with three intersections

When using the Vrigen control method, the three intersections as indicated in Figure 4.14 are considered. The intersection Basisweg-A10 ramp is considered to be one big intersection. Vrigen supports coordination for an intersection this size. The three intersections operate independent with respect to each other. There is no coordination between them, only detector information is available. The detectors are inductive loops as found at traffic lights in general (see Figure 4.15).

To make the comparison with the other methods as good as possible, all clearance times are set to 2 seconds, just like in the S-model. Also, the control schemes are the same as for the MPC controller, see section 4.4.3. The maximum cycle time for all three intersections is set at 90 seconds. This is done, because the middle intersection (Basisweg-A10 ramp) has a calculated cycle time of 77 seconds in case of peak hour demand pattern as described in Section 3.5.1. So the control freedom with respect to the other two control methods is a cycle time that is up to 13s shorter (up to 17% shorter cycle time). This might look small but since all the clearance times are set to 2s, the internal lost time is only 8 seconds (2s for each phase). This can be considered short for a regular 4-phase controlled intersection. When the internal lost time is higher, relatively more time of the total cycle time is lost. In that case a higher cycle time could be beneficial. In this case, a lower cycle time yields 20% reduction in lower average waiting times when compared to a cycle time of 90s.

Figure 4.15 – Inductive loop (detector)

4.6 Fixed time control

Fixed time control means traffic schemes are generated in advance. Based on the expected traffic demand, green times are assigned to each signal group. On the street, the schemes will be executed exactly as generated. There is no flexibility with the green times, nor will they react to the presence or absence of vehicles.

An advantage of fixed time control is that coordination between separate intersections can be implemented relatively easy by aligning the green times of large traffic streams. To do this, the same traffic schemes from Vrigen are used, as for the Vrigen control method. This is to determine the sequence of the phases. Then, the green times are determined in Cocon.

To achieve coordination between intersections with a fixed time control method, it is necessary that the relevant intersections (in this case all three of them) have similar cycle times or have multiples of each other's cycle time. For the chosen traffic load patterns, the intersection Basisweg-A10 ramp needs the highest cycle time. The other two intersections can handle the traffic with a lower cycle time. In this case, it is chosen not to do so, to benefit from coordination. There are various points of view on the question which method provides the highest traffic performance. As opposed to the coordination between intersections, Vrigen's vision is that coordination between intersections is not relevant; the highest performance will be reached when all intersections are locally optimized. For the chosen traffic load patterns, experiment 1 (see section 3.5.1) will provide an answer.

4.7 Software used by traffic control methods

Now the three control methods used in this thesis are discussed, this section summarizes the software necessary for each control method. The necessary software is indicated in Table 4.8.

Table 4.8 – Software used for the different control methods. "x" indicates necessary software, "o" indicates the optional use of software

	Fixed time control	Vrigen control	MPC
Matlab	o	o	x
Vissim simulator	x	x	x
fixed time controller	x		
Vrigen	o	x	x
Trafcod		x	x
S-model			x
Optimizer			x

This table indicates the necessary use of software (x) and optional use of software (o). The traffic simulator Vissim will be discussed in Section 4.8. In cases of fixed time control and Vrigen control, Matlab is optional. The use of Matlab is not a necessary for these control methods. However, all simulations with these control methods were executed with Matlab, because with Matlab the process of running multiple runs can be automized. Without Matlab, every run should start by hand which is rather time consuming. The optional use of Vrigen for fixed time control is to derive traffic control schemes. Another option to derive those schemes is by using the software package Cocon or simply by hand.

4.8 Microscopic simulator (Vissim)

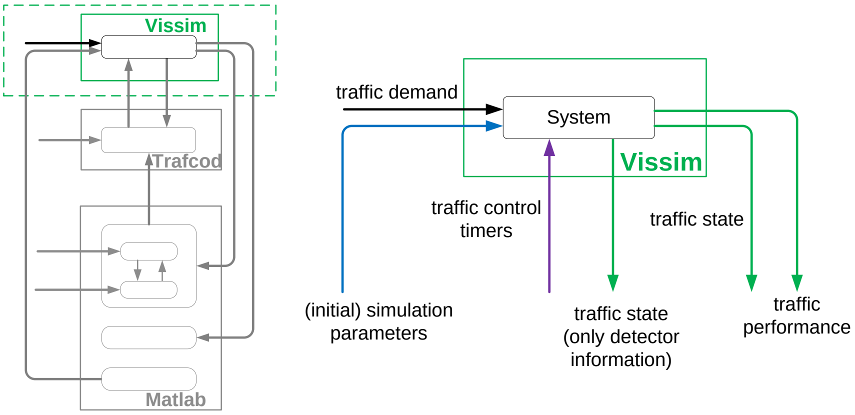


Figure 4.16 – Position of Vissim in the simulation environment

4.8.1 Different options

There are four microscopic models that could be considered: Fosim, Vissim Paramics and Aimsun.

Fosim

Fosim is a microscopic model calibrated for Dutch freeways. This model does not support traffic lights and is therefore not suitable.

Aimsun

A macro-, meso- and microscopic model are combined in Aimsun. Interfaces with other software are available and custom software written in for example C++ can be implemented. The use of traffic lights is supported.

Paramics

This microscopic model offers limited support for interfaces with external software. The use of traffic lights is supported.

Vissim

This microscopic model offers an interface to external software by activeX- or COM-server. The use of traffic lights is supported.

For all micro simulators, software licences are available. Based on the possibilities for external software support, Aimsun and Vissim are both suitable. Since the author is already familiar with Vissim, that is the obvious choice for selecting a microscopic model.

4.8.2 The network in Vissim

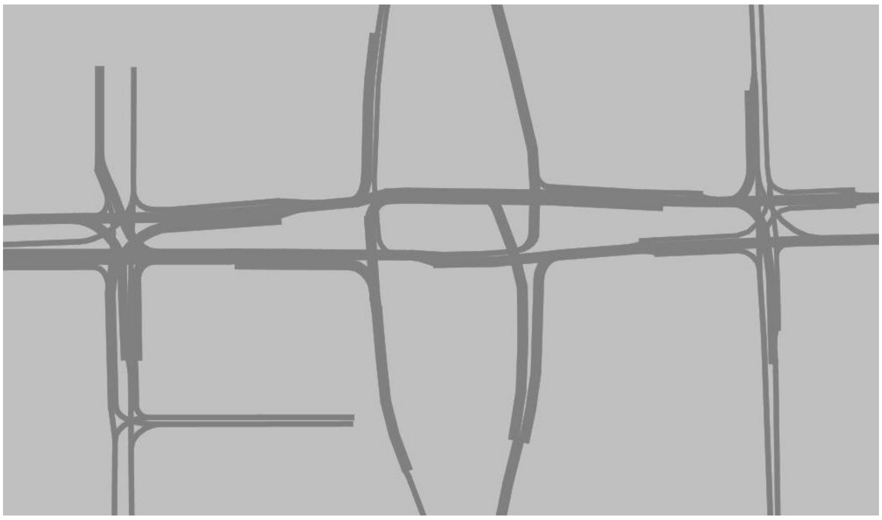


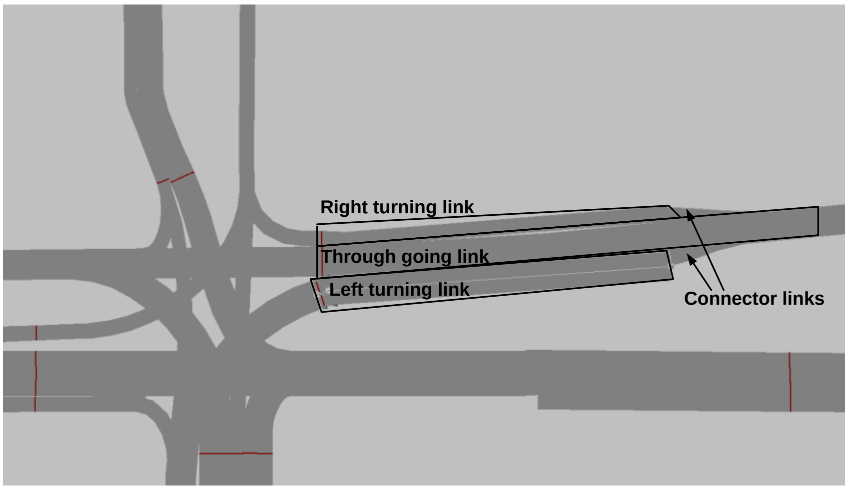
Figure 4.17 – Screenshot of the S102 network in Vissim

The network in Vissim (Figure 4.17) is built according to the real road geometry. Because of the traffic load patterns, it was necessary to extend some left turning and right turning lanes. This was done to prevent disproportional blocking of the road. When both the traffic demand and the cycle time are known, the average number of arriving vehicles per lane can be calculated. If the number of arriving vehicles is higher than there is room in the turning lanes, blocking of other vehicles can be expected. In case the turning lanes would not be extended, the network lay-out would have an unwanted effect on the traffic performance. One sub goal of this thesis is to investigate the traffic performance of different control methods and that should not be influenced by other factors.

At locations where lane pre-selection occurs, it is chosen to separate the selector links relatively early (see Figure 4.18). Vehicles can only enter the left and right turning links by the connector links. This prevents vehicles from changing lanes at the last instant. In reality, vehicles are able to change lanes at the last instant. However, in Vissim the frequency of this phenomenon is high. It has a serious influence in decreasing the traffic performance, because when this happens, upstream traffic is blocked. Therefore, forced early pre-selection of lanes is chosen because it is believed that this way of modelling yields more realistic results.

To detect vehicles in front of a traffic signal (both waiting and approaching), the data of all vehicles can be extracted. This includes vehicle ID, x position, y position, speed, link number, etcetera. The links in the Vissim-model are chosen in such a way that filtering by link number is sufficient for vehicle detection in front of a traffic light for left turning, right turning and through going traffic (see Figure 4.18). A distinction is made between a waiting and driving vehicles. Vehicles are considered waiting if their speed is lower than 5 km/h.

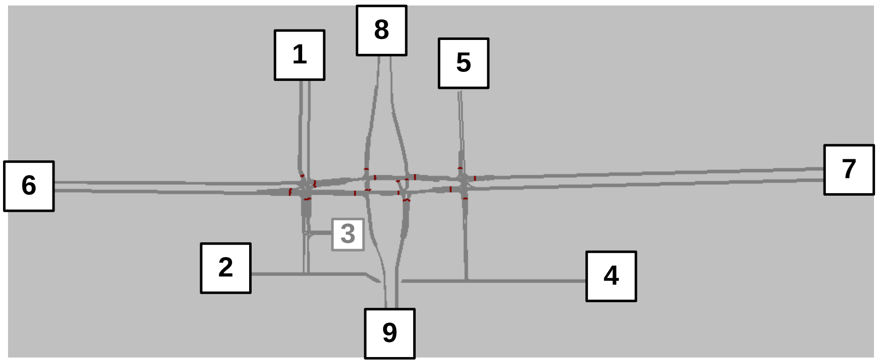
Figure 4.18 – Physical separation of pre-selection links



4.8.3 OD matrix

The traffic load pattern is put in Vissim by using the dynamic traffic assignment. To do so, it is required to define zones which are origins as well as destinations. In the S102 network, 9 zones are defined (see Figure 4.19).

Figure 4.19 – 9 OD sources. Source number 3 is not used.



In case multiple routes are available, the dynamics assignment will deal with the route choice. Vissim is known for the possibility to yield strange results when it comes to route choice. To eliminate this possibility, the Hatostraat is cut in two parts. In this way, the network offers only one possible route per OD pair.

Text files (*.fma files) contain the OD matrix and the load factor. The load factor multiplies all productions and attractions to obtain the applied traffic loads. So it depends on the length of the intervals how many fma files are needed. In this thesis, the choice is to run a simulation for 2,5 hours. Every 15 minutes, the load factor should be able to change. This means 10 fma files are needed.

The traffic load patterns are stored in the file flows_turnrates.xls. The function createFma.m in Matlab creates the necessary 10 fma files.

4.8.4 Random seed

Vissim is a stochastic model. The traffic load pattern that Vissim will generate during a simulation is not exactly equal to the given values of the OD matrix. Vissim applies some variation itself. This variation is random, but this randomness it determined by the value of the random seed. Random seed is any number between 1 and 99999. When performing multiple simulations with the same random seed, they should all generate the same traffic pattern.

For all experiments, the same set of random seeds is chosen. It is excluded that a different traffic performance is the result of a different random generated traffic pattern. The random seeds have values ranging from 15 to 19. Every run has a different value for the random seed, there are 5 runs in total.

4.8.5 Ramp metering

For experiment 2 (see section 3.5.3), ramp metering is applied in Vissim when there is a desired maximum outflow to both the A10 (northbound) and the S102 (eastbound). This is done by using a regular traffic light with fixed time control with a cycle time:

$$T_c = \frac{3600}{Q_{out,ref}}$$

The green signal is shown for one second, followed by a 3 second amber phase. Every cycle time, one vehicle is allowed to enter the highway or the adjacent subnetwork.

4.9 Comparison Vissim and S-model

When MPC is used, the optimized control signal is obtained by using the S-model. It is important to test the compatibility between both Vissim and the S-model. The input variables of the S-model are green times, the output is TTS. For Vissim, the same can be done. By using the fixed time controller in Vissim and making a run with the same simulation time as the S-model. Vissim can provide many values as output, TTS is just one of them.

This comparison can be seen as a sensitivity analysis. The method of comparison is to have four different traffic loading schemes in which the demand in constant for the total simulation time. All traffic loading schemes will be subject to a combination of 851 different traffic control timers a shown in Table 4.11. The choice is made to only use two variables: i and j. This is because with two variables, it is possible to plot a (3D) graph and inspect the values of the objective function (TTS) visually. When using more variables, it is difficult to interpret the result since visualizing the data is hard.

4.9.1 Control variables

Table 4.9 – Phase sequence for Basisweg-A10 (northbound)

Phase	Signal groups green
1	1, 2, 8
2	1, 4, 6
3	8, 9

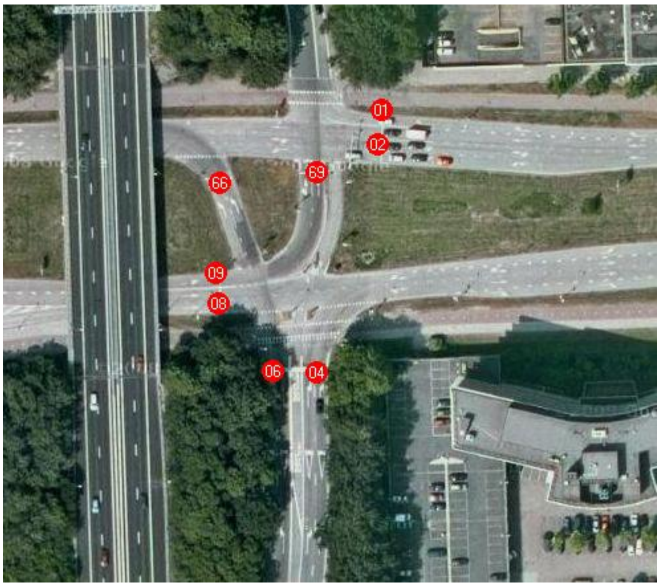


Figure 4.20 – Signal groups at intersection Basisweg-A10 (northbound)

Table 4.10 – Phase sequence for Kabelweg-Transformatorweg

Phase	Signal groups green
1	1, 10, 11
2	7, 8, 9, 10
3	1, 2, 3
4	5, 6, 7

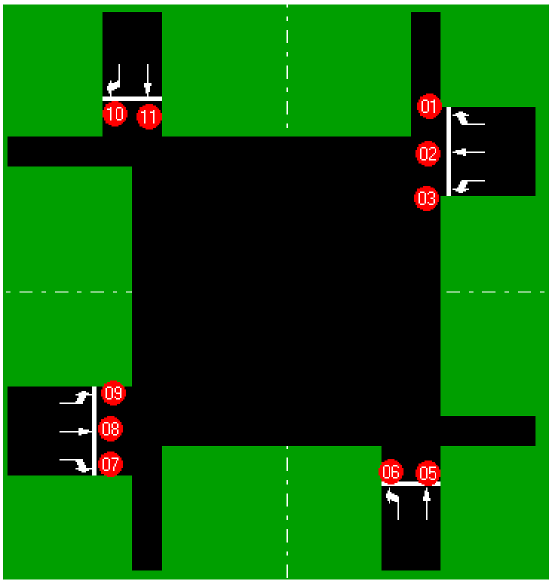


Figure 4.21 – Signal groups at intersection Kabelweg-Transformatorweg

Table 4.11 – Traffic control timers for S-model/Vissim testing. Two variables are present: i (ranges from 1 till 37) and j (ranges from 1 till 23)

Intersection	Phase 1	Phase 2	Phase 3	Phase 4
Basisweg-Laguardiaweg	22	22	23	23
Basisweg-A10 (southbound)	30	30	30	
Basisweg-A10 (northbound)	**11+i**	**49-i**	30	
Kabelweg-Transformatorweg	22	**35-j**	**11+j**	22

The variable i runs from 1 to 37. For the intersection Basisweg-A10 (northbound), phases 1 and 2 will be varied (see Table 4.11). The minimum green time is 12s and the maximum green time is 48s. The 3rd phase has a fixed green time. The traffic load is chosen in such a way that no congestion will occur for directions 8 and 9 (see Figure 4.20) with a fixed green time of 30s. If congestion does occur, one can be sure the cause is variable j; a short green time in phase 2 at Kabelweg-Transformatorweg.

The variable j runs from 1 till 23. For the intersection Kabelweg-Transformatorweg, phases 2 and 3 will be varied. These phases have a minimum green time of 12s and a maximum green time of 34s. Phases 1 and 4 with a fixed duration of 22s and the traffic load is chosen in such a way that directions 5, 6, 10 and 11 will not experience congestion due to a shortage of green time at this intersection. If congestion occurs, one can be sure that the congestion is caused by a change in variable i. The same reasoning also applies to the two intersections that have only fixed time control; none of their directions will experience congestion because a shortage of green time at their respective intersections. However, they might experience congestion because of changes in variables i and j.

4.9.2 Traffic load patterns

The same set of control variables i and j as described in Section 4.9.1, will be subject to four different loading patterns, shown in Figure 4.22, Figure 4.23, Figure 4.24 and Figure 4.25. In all figures, only the flows entering the network are given to keep them compact and easy to understand. The same turning rates apply as described in Section 3.5.1.

Pattern 1 (Figure 4.22) contains the same entry flows (600 veh/hour) at all points in the network. This can be considered non-busy traffic. Pattern 2 (Figure 4.23) represents a peak hour pattern with double loads on the main directions. Pattern 3 (Figure 4.24) also represents peak hour traffic. The differences with pattern 2 are higher loads on the main directions that contain westbound traffic. This is done to investigate the propagation of delay caused by phase 2 in intersection Kabelweg-Transformatorweg. The fourth pattern (Figure 4.25) represents event traffic.

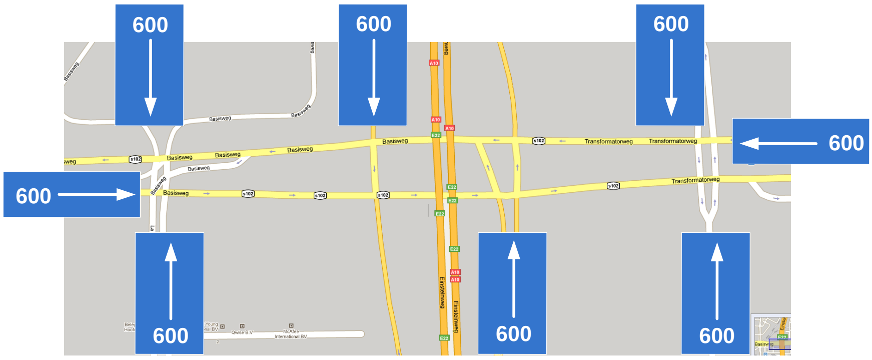


Figure 4.22 – Traffic load pattern 1. Flows are in vehicles/hour.

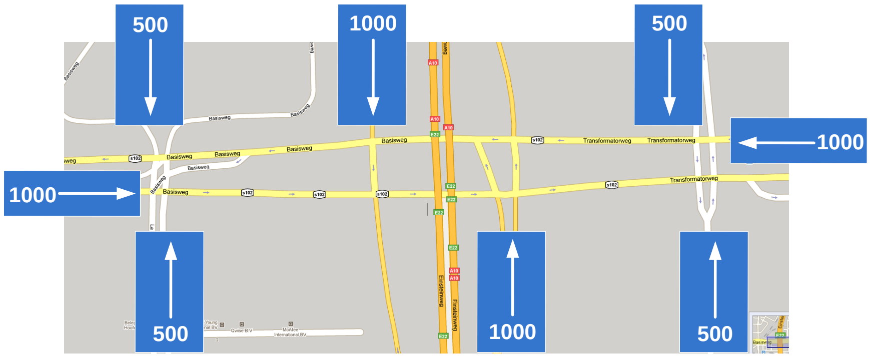


Figure 4.23 – Traffic load pattern 2. Flows are in vehicles/hour.

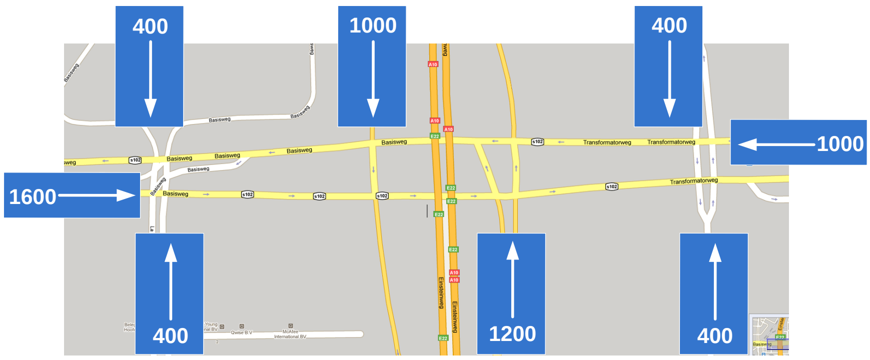


Figure 4.24 – Traffic load pattern 3. Flows are in vehicles/hour.

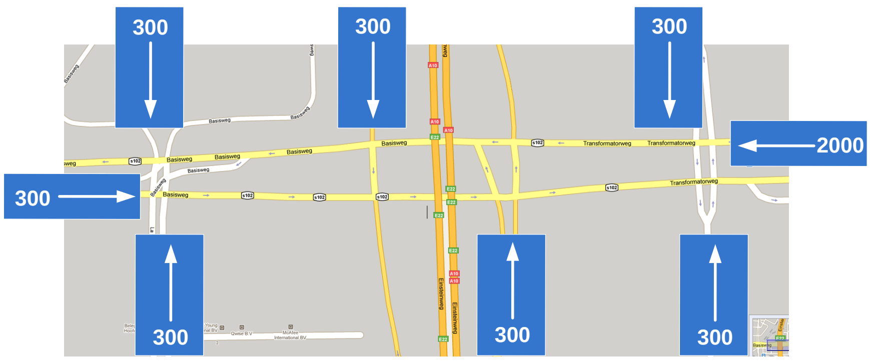
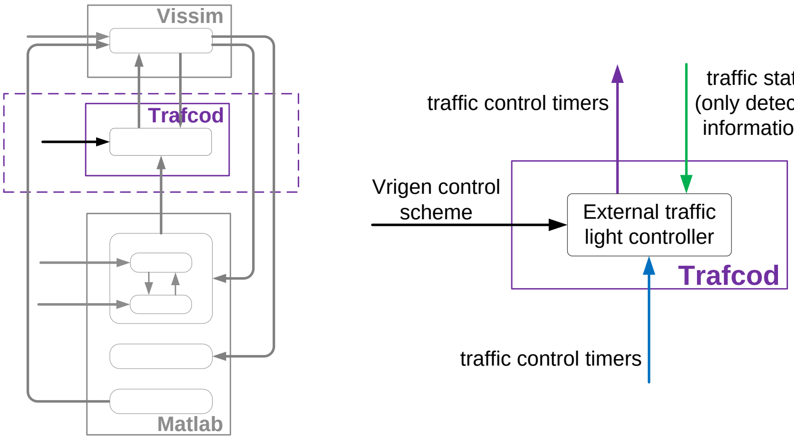


Figure 4.25 – Traffic load pattern 4. Flows are in vehicles/hour.

4.10 Trafcod as a traffic controller for MPC

Figure 4.26 - Position of Trafcod in the simulation environment



Direct communication between Matlab and Vissim exists. For traffic controllers other than a fixed time controller, Vissim requires an external controller.

PTV (Vissim's developer) has several options for external controllers. Since the licenses are not available, another option is chosen. The external traffic controller Trafcod is used, which was developed at Delft University of Technology. Trafcod uses control schemes generated by Vrigen and executes according to the Vrigen timers.

Trafcod was adapted in such a way, that the timers of Vrigen can be overwritten by custom values. This is done by text files (*.adj files) with the same name as the Vrigen control scheme. Ten times per second, Trafcod checks for new *.adj files. When found, the timers are implemented and the files are deleted.

Timers that can be changed include for example fixed green time, extension green time and amber time. The amber time is always constant (3s) and should not be changed. The extension green time gives a maximum green time, which will be realized only when traffic is detected during the green phase. Using this option would mean it is not possible to maintain a constant cycle time. In case of an MPC controller, Trafcod should just execute the calculated green times. Therefore the fixed green time should be used.

Executing fixed green times only implies that Trafcod is only a way to give the green times to Vissim. This is indeed true, so the feedback from Vissim to Tracod (the detector information) as indicated in Figure 4.26 is not used. For the proper functioning of Trafcod, the feedback link with Vissim should be maintained.

In *.adj files, the timers can be adjusted by first calling the kind of time (fixed green time [TGF], extension green time [TGX]) and its signal group as defined in the Vrigen control scheme. Next to this code, the desired timer should be presented in tenths of seconds. A typical *.adj file looks like this:

```
TGF01  179
TGF02  179
TGF03  179
TGF04  179
TGF05  178
TGF06  169
TGF07  169
TGF08  169
TGF09  169
TGF11  178
TGF12  169
```

The *.adj files are created with the function adjustTimersS102.m in Matlab.

Please note that this section has nothing to do with the Vrigen control method. In case of the Vrigen control method, (see Section 4.5), Trafcod will work on its own without being influenced by Matlab.

4.11 Number of runs

The number of runs can be calculated with the following formula (Muller, 2004):

$$n \geq t^2_{\frac{1}{2}\alpha,n-1}\left(1+\frac{1}{2}\xi^2\right)\frac{X_s'^2}{X_d^2} \qquad \textbf{Eq. 4.4}$$

In which

X_s' = the sample standard deviation

X_d = the accepted deviation

α = the reliability

ξ =the abscise of the normal distribution attain ability value

t = the value from the student-distribution

To determine the number of runs necessary (the amount of repeating simulations with a different random seed), first a reliability percentage should be selected. However, due to limited time, 5 runs per experiment can be made. In a reverse way, the reliability can be determined.

Ten short runs of 900s have been made with the MPC controller in Vissim. The performance indicator is the travel time. With the reliabilities fixed at 90% of 95%, the accepted deviation can be calculated. The result is in Table 4.12. The data can be found in Appendix D.

Table 4.12 – Determining the acceptable deviation

	95%	90%
Mean (of the samples)	3.234	3.234
St. Dev (of the samples)	597	597
Acceptable St dev	741	494
T value	1,812	1,372
N	5,0	5,0

5. Results and discussion

In this chapter, the results for both the sensitivity analysis between the S-model and Vissim (Section 5.1) and the experiments (Section 5.2) will be presented and interpreted.

5.1 Sensitivity analysis between S-Model and Vissim

In this section the results of the sensitivity analysis of the comparison between the S-model and Vissim are presented. There are four traffic loading patterns. For each pattern, at first the traffic load pattern will be repeated for convenience. The total time spent in the network (TTS) will be given in the form of a 3d graph for both the S-model and the Vissim model. This graph indicates the TTS (in vehicles*hours) for each value for i and j. Both graphs will have the same scaling and viewing angles so comparing is easy. The colorbar next to the graph indicates the minimum and maximum value of the TTS.

To remind the reader about the set-up of this experiment (see Section 4.9), the phases which are influenced are repeated once again. The lay-out of the network is given in Figure 5.1 . The variable i influences the duration of phase 1 and 2 of intersection Basisweg-A10 (northbound), see Figure 5.2. The variable j influences the duration of phases 2 and 3 of intersection Kabelweg-Transformatorweg, see Figure 5.3.

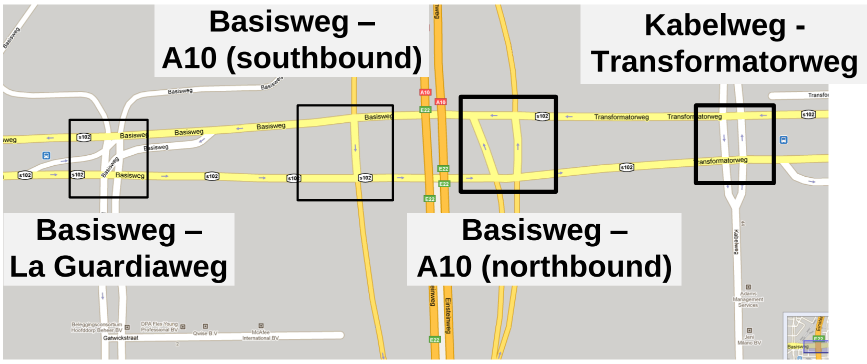


Figure 5.1 – Map with the four intersections of the S102 network

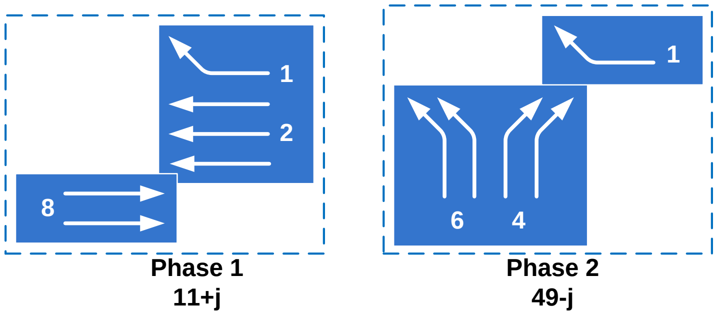


Figure 5.2 – Signal groups at intersection Basisweg-A10 (northbound) visualized with arrows. The duration per phase is indicated in seconds.

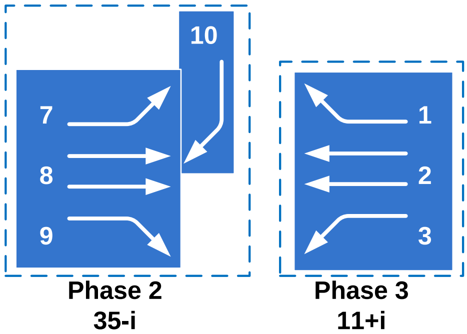


Figure 5.3 – Signal groups at intersection Kabelweg-Transformatorweg visualized with arrows. The duration per phase is indicated in seconds.

Table 5.1 – Traffic control timers (in seconds) for S-model/Vissim testing. Two variables are present: i (ranges from 1 till 37) and j (ranges from 1 till 23)

Intersection	Phase 1 (s)	Phase 2 (s)	Phase 3 (s)	Phase 4 (s)
Basisweg-Laguardiaweg	22	22	23	23
Basisweg-A10 (southbound)	30	30	30	
Basisweg-A10 (northbound)	11+j	49-j	30	
Kabelweg-Transformatorweg	22	35-i	11+i	22

5.1.1 Traffic load pattern 1

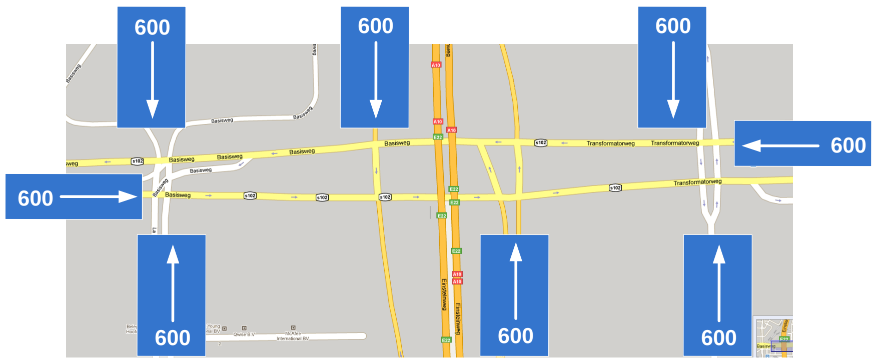


Figure 5.4 – Traffic load pattern 1. Flows are in vehicles/hour

Figure 5.5 – Total Time Spent for the Vissim model, loaded with traffic pattern 1

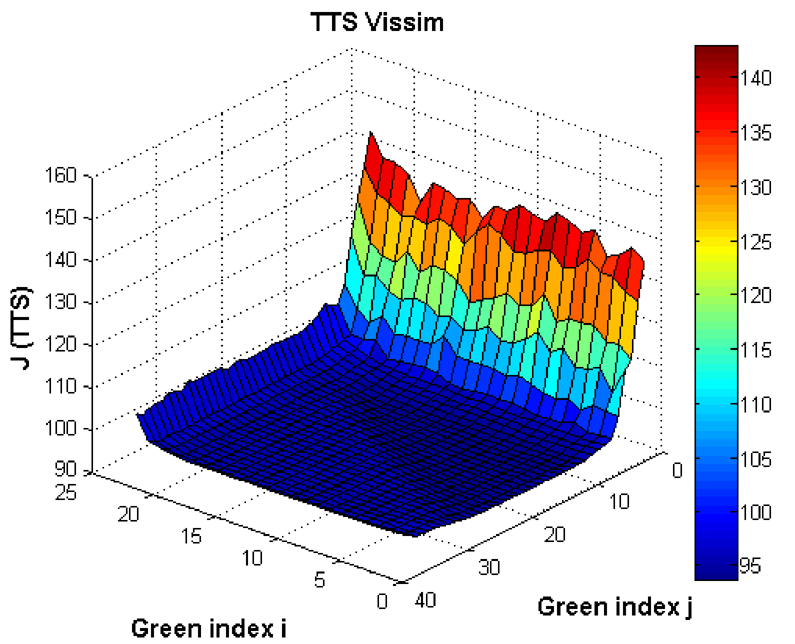
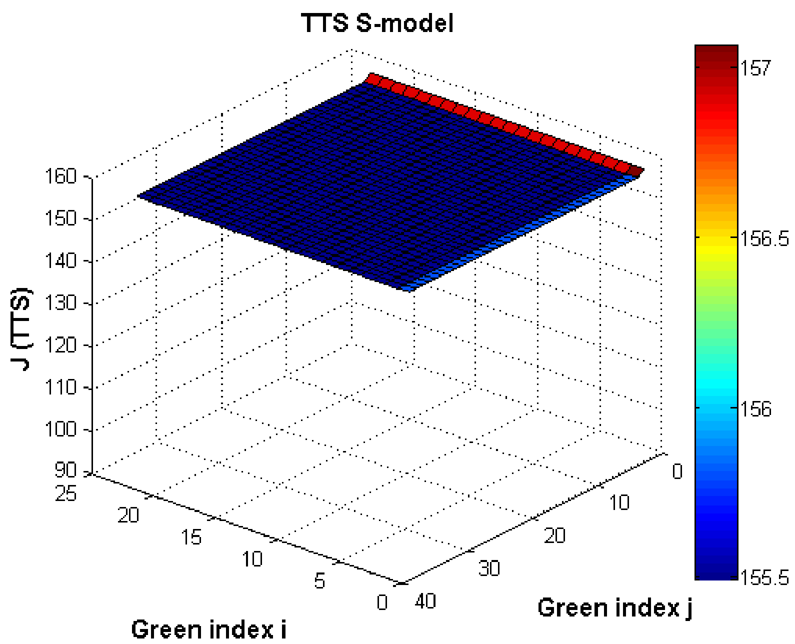


Figure 5.6 - Total Time Spent for the S-model, loaded with traffic pattern 1



The first traffic pattern is one with similar flows from all directions (Figure 4.22). For both cases, it can be seen that the value of TTS is more or less constant, except at the edges. The values of TTS for the S-model are much higher (155) when compared to the TTS of Vissim (95) (see Figure 5.5 and Figure 5.6). For the sensitivity of the model, this does not have to be an issue since the optimal control signal is about minimizing TTS. If the values are significantly different, but the lowest values for TTS are reached at the same values of i and j, the compatibility between the S-model and Vissim can be considered excellent.

The differences are at the edges. At the lowest value for j (j=1), both models indicate an increase of TTS. However, the change of TTS in the Vissim model is much higher: +52% compared to the lowest value while for the S-model this increase is just +1%. An increase this small could be considered merely negligible.

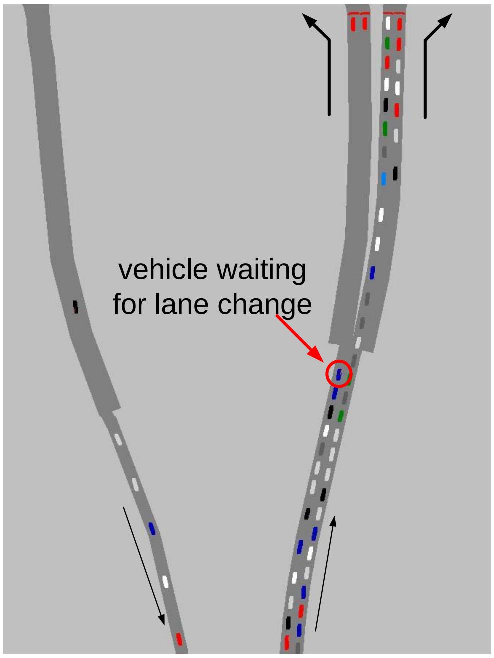


Figure 5.7 – Inefficient use of the turning lanes because of blocking. The vehicle in the red circle needs to go to the right and waits for a gap.

The rapid increase in TTS for Vissim starts at j=6 until j=1, which corresponds to a green time of 17s (and down to 12s) for phase 1 (see Figure 5.2). For the S-model, this increase starts at j=2. When looking at the traffic stream present for directions 1 and 2 of phase 1, the traffic load is 273veh/h per lane. (This can be derived from the loading pattern and the turning rates). When assuming a cycle time of 90s and a saturation flow of 1800 veh/h, the minimum green time would be: 273*(3600/1800)*(90/3600) = 13.65s.
When a shorter green time is given, one would expect queues to build up. Vissim indicates a green time of 17s is necessary to prevent queues from building up. The difference of 4s can be explained by the phenomenon of blocking (see Figure 5.7). In situations where inflow equals outflow, and where the turning rates are exactly 50%(left)-50%(right) all the time, blocking will not happen. However, in reality (as well as in Vissim) the turning rates are never equally distributed in time. In case there is a higher demand for one of the directions (in the case of Figure 5.7, there is more demand for right turning traffic), this traffic might block other traffic. In Figure 5.7, the left turning traffic is severely blocked and cannot enter its lane. A blocking vehicle needs some time to resolve. On average for this case, the time needed to resolve blocking is around 4 seconds.

The small increase in TTS of only 1% in the S-model (Figure 5.6) needs to be investigated. This increase happens at j=1. No change in TTS means the variables i and j have no influence; hence the smallest green times are sufficient to process the traffic. This is strange, because the is the theoretical minimum green time was calculated to be 14s (which can only be achieved in an ideal case that is unlikely to occur) while the lowest green time of 12s in the S-model still processes the traffic without an increasing TTS. Suspicion rises that the flows in the network are lower than expected.

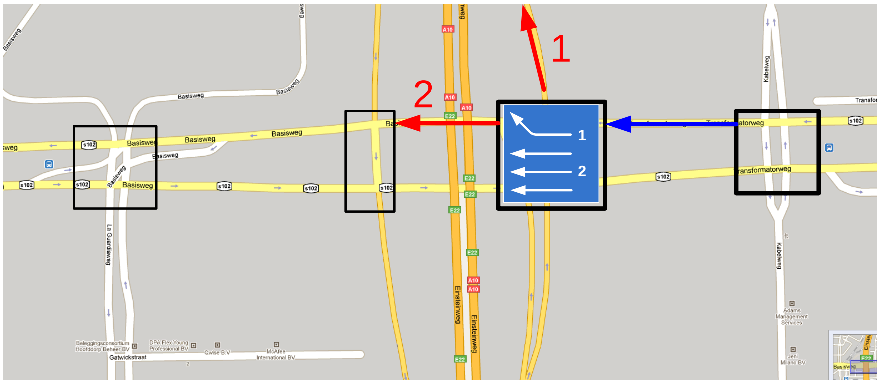


Figure 5.8 – Indicated with a red arrows are the link upstream of intersection Basisweg-A10 (northbound). The blue arrow is the feeder link.

Consider intersection Basisweg-A10 (northbound) in Figure 5.8. In the S-model, the amount of vehicles that can leave a link depend on three factors:

1. Amount of cars present in the link
2. Maximum saturation flow
3. Available space in the upstream link

A negative contribution from factors 1 and 2 is not possible, since the entry flows and the saturation flows are well defined in the S-model. It means on a structural basis, the available space in the upstream link is not sufficient. Applied to the example in Figure 5.8, it means not all vehicles that want to leave the blue link can enter the red links. Apparently, the downstream links (in red) are congested or do not offer sufficient space. For direction 1, the downstream link is the on-ramp to the A10 motorway, and this link is set to infinite capacity, so all vehicles that want to enter the A10 are able to do so. Link 2 contains limited space for vehicles, perhaps the link cannot accommodate all vehicles that want to enter. This principle can be applied to all links in this corridor.

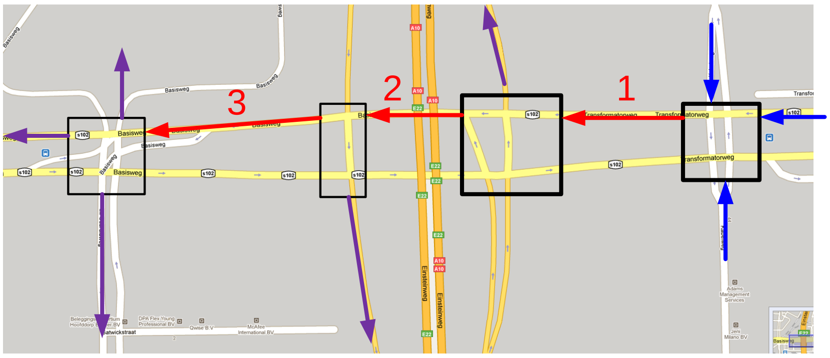


Figure 5.9 – Indicated with a red arrows are the link with limited space, purple links have infinite space. The blue arrows are links at which vehicles enter the network.

In Figure 5.9, we start at intersection Kabelweg-Transformatorweg. The blue arrows indicate the three entry flows that generate westbound traffic. The first upstream link is red link 1. This link has limited space; as many vehicles can enter as there is space in this link. Next, this link 1 now is the entry flow for two links, the A10 on-ramp (in purple) and red link 2. The purple link has infinite space, so all the vehicles that want to enter, are able to do so (in this case, we do not consider ramp metering). The limitation could be red link 2, which has limited space. The downstream links for link 2 are the other on-ramp to the A10 motorway (southbound) and red link 3. Here it is the same story; only link 3 could be a limitation. The links leaving the network have infinite space and do not limit the outflow. So, one of the three red links determines the maximum flow for this whole corridor. When the demand is higher, queuing will occur at the three blue arrows.

This finding is confirmed when consulting the additional output of the S-model; the outflows are smaller than the inflows, so queues are growing larger at the links entering the network. This should not be the case, because queuing should only occur roughly at the same values for i and j when compared to queues building up in Vissim. It explains why the value of TTS for the S-model is much higher compared to TTS of Vissim; growing queues means that the travel time becomes larger

The next question is then, why is the space in one of the three red links (Figure 5.9) limited in such a way that it cannot handle the relatively non-busy traffic load (see Figure 4.22)? This issue has two causes:

- The time step of the S-model
- The way free space in a link is assigned to different directions

Time step of the S-model

The S-model is a macroscopic traffic model. For correct traffic modelling, it is important that the time step is shorter than the free flow travel time on the shortest link. The time step is equal to a cycle time (90s). Assuming a free flow speed of 50km/h, one time step is equivalent to a travelled distance of 1250m. The shortest link in the S102 network is 90m, so this condition is not met.

The consequence for the S102 network is the following. Suppose a link has space for 30 vehicles to queue. During a cycle, 50 vehicles want to enter. Even though the green time is large enough process this amount of vehicles, this will not happen since only 30 vehicles can enter the link. The other 20 vehicles will have to wait in the upstream link.

Two things can be done to solve this issue:

- Choose a smaller time step
- Increase the length of the links

The philosophy of the S-model is to have one cycle time (or multiple cycle times) as one time step. This only leaves the option to increase link lengths. The links would have to be increased to such an extent that one could wonder if coordination is needed when distances between traffic lights are bigger than 1km. The S102 network was chosen because it represents a typical road network lay-out in The Netherlands, where controlled intersections are at close distance with respect to others.

When this issue was detected, insufficient time was available to come up with a solution. Therefore, recommendations will be given in Section 6.5.1.

Free space assignment in the S-model

In the S-model, the free space in a link is assigned according to the turning rates of the incoming link. See Figure 5.10 for an example. In this example, an empty downstream link is assumed. Traffic from three different directions wants to enter. This traffic is indicated by a green, a blue and a purple arrow. The available percentage of the free space that will be assigned to either of them will be according to the turning rates of the lane lay-out in front of the traffic signal. In this example, all lay-outs are the same: 1 left turning lane, 1 right turning lanes and 2 through going lanes. Another example of a matching lane lay-out is given in Figure 5.11.

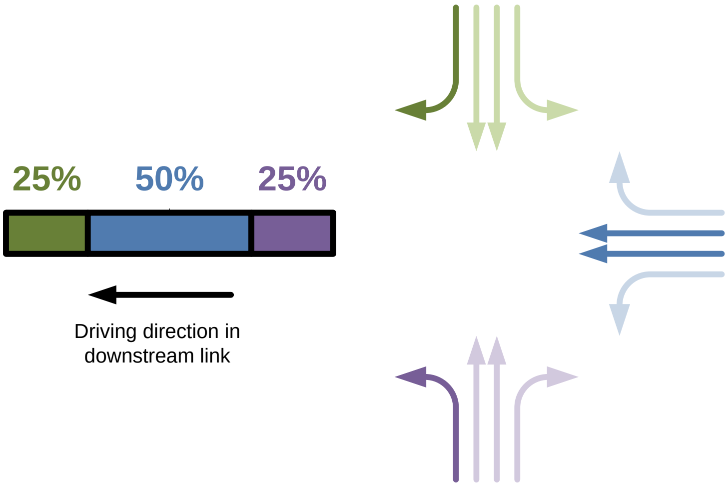


Figure 5.10 – Assignment of free space in an empty link. Lane lay-out at stop lines is similar.

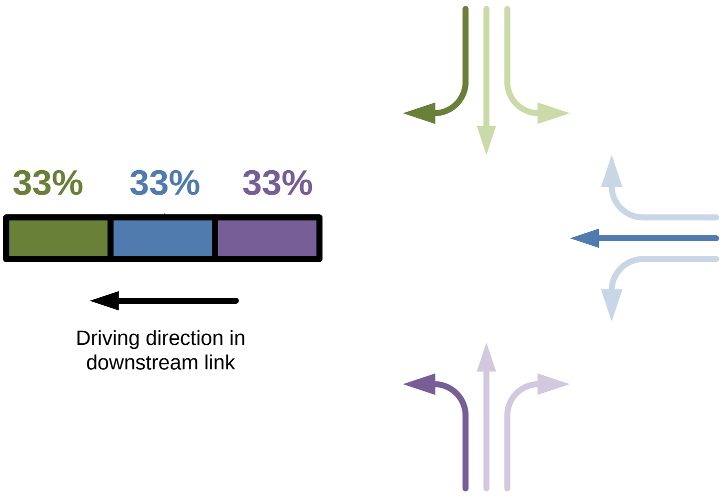


Figure 5.11 – Assignment of free space in an empty link. Lane lay-out at stop lines is similar.

There are two ways in which the available space in a downstream link is not assigned in a realistic way. The first one is when the different directions have different lay-outs in turning lanes in front of a traffic light. An example is given in Figure 5.12. In this case, the total of all turning rates adds up to more than 100%, namely 141%. The free space is overestimated in this case. This means the S-model might predict an optimal control signal in which the amount of vehicles entering the link is 42% more than the link can handle. Other directions at the intersection will be blocked to and therefore perform worse. Underestimation of the free space is also possible with other lane lay-outs, see Figure 5.13. Then, the free space is used inefficiently.

Figure 5.12 – Assignment of free space in an empty link. Lane lay-out at stop lines is different. Free space in the link is overestimated (by 42%).

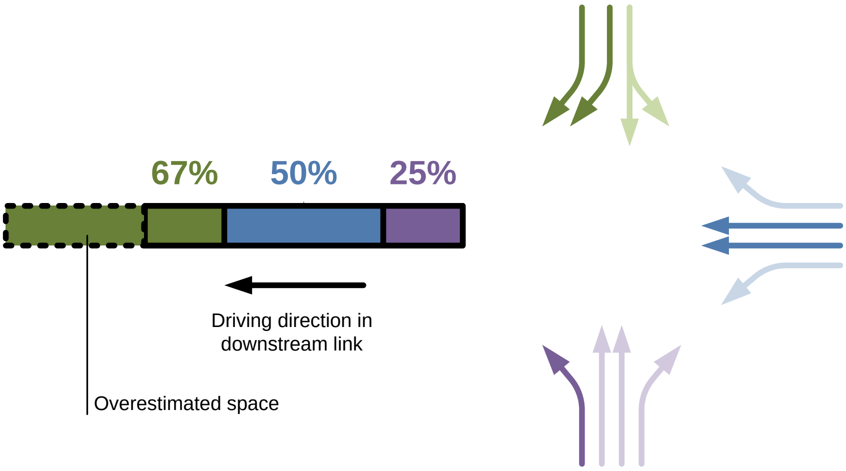
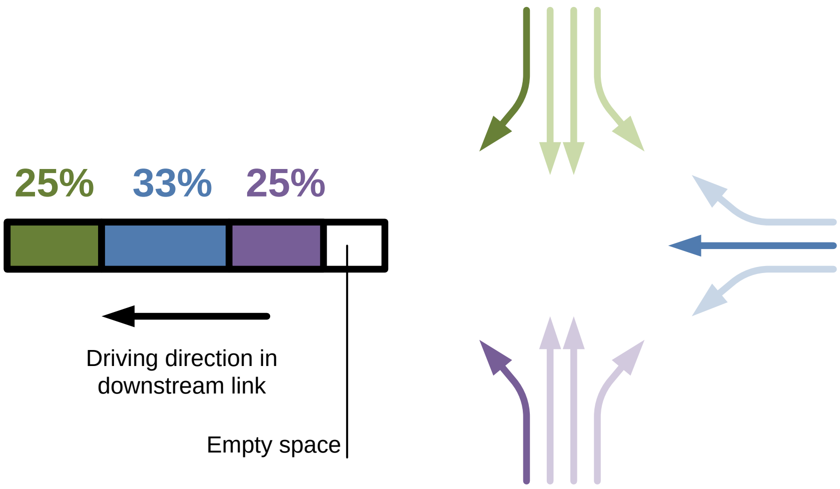


Figure 5.13 – Assignment of free space in an empty link. Lane lay-out at stop lines is different. Free space in the link is underestimated (by 17%).



The second mistake that can be made is a mismatch in flow rates. The current design of the S-model program needs similar flows from all directions to fully use the free space available. When looking at Figure 5.10, suppose the total free space in the link is 1100 vehicles. In Figure 5.14, the green and purple arrows both have a flow of 10veh/h and the blue arrow has a flow of 1000veh/h. For both the green and purple arrow, 275 vehicles can enter (only 10 actually do) while for the blue arrow 550 vehicles can enter. Although enough space is available to handle all vehicles, only 56% of the total amount of vehicles can enter. And 44% of the free space is not used.

This issue can be solved in the S-model program. Time to implement changes to the program however was not available.

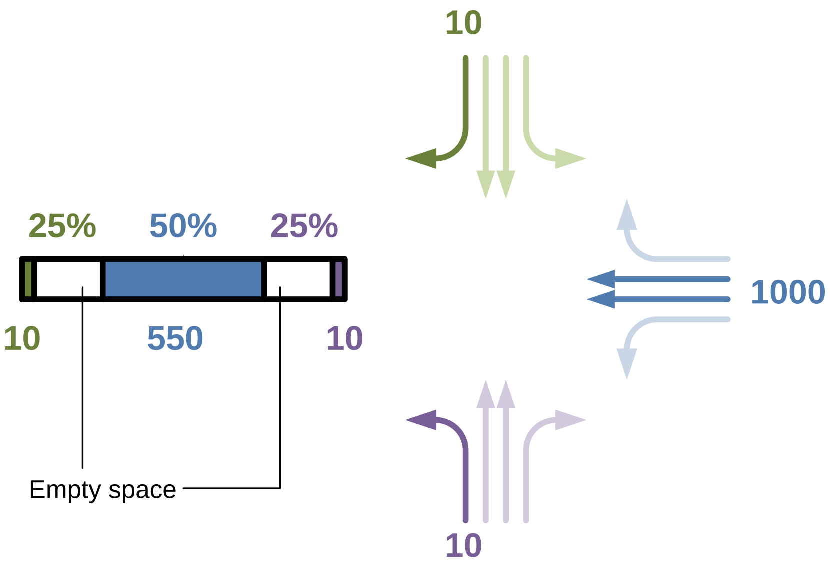


Figure 5.14 – Assignment of free space in an empty link with space for 1100 vehicles. Entry flows are not similar. Only free space according to turning rates can be used.

5.1.2 Traffic load pattern 2

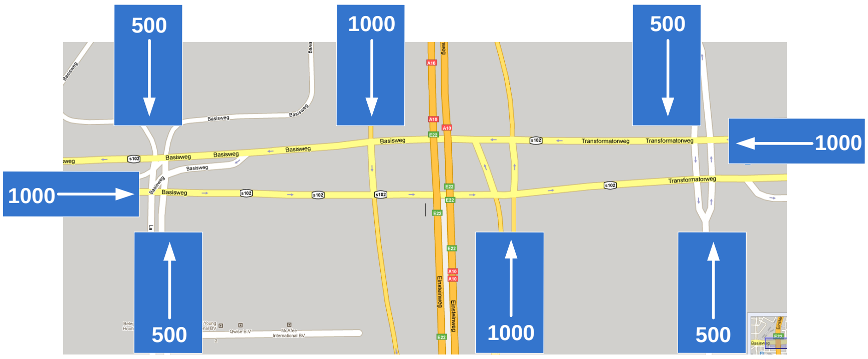


Figure 5.15 – Traffic load pattern 2. Flows are in vehicles/hour.

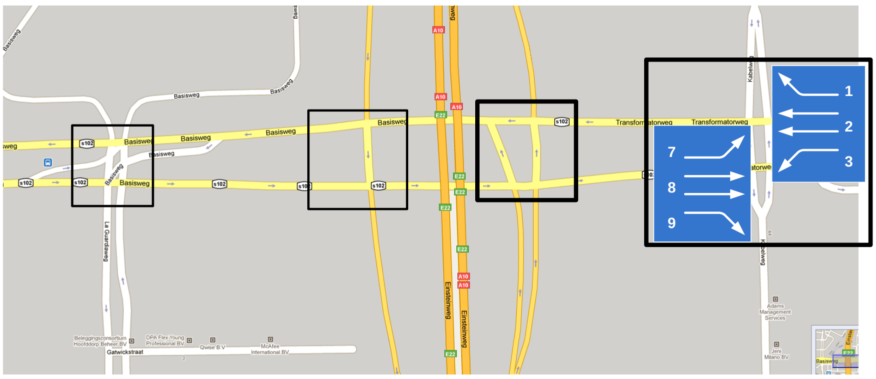


Figure 5.16 – Critical directions for values of i at intersection Kabelweg-Transformatorweg.

The second When looking to the result in Vissim, the TTS starts increasing sharply at i=3 (to i=1), i=18 (to i=23), j=8 (to j=1) and j=33 (to j=37). Just like in the case of pattern 1, it should be checked if these values can be expected. First for i=3, this yield a relatively small green time for directions 1, 2 and 3 (Figure 5.16). The flow present there is 250veh/h per lane. The expected minimum green time would then be: 250*(3600/1800)*(90/3600) = 12.5s.

Figure 5.17 – Total Time Spent for the Vissim model, loaded with traffic pattern 2

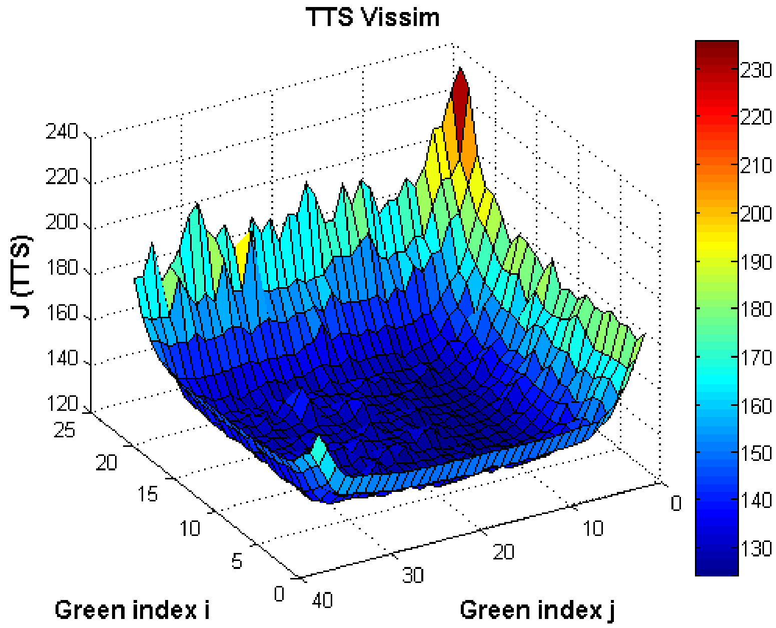
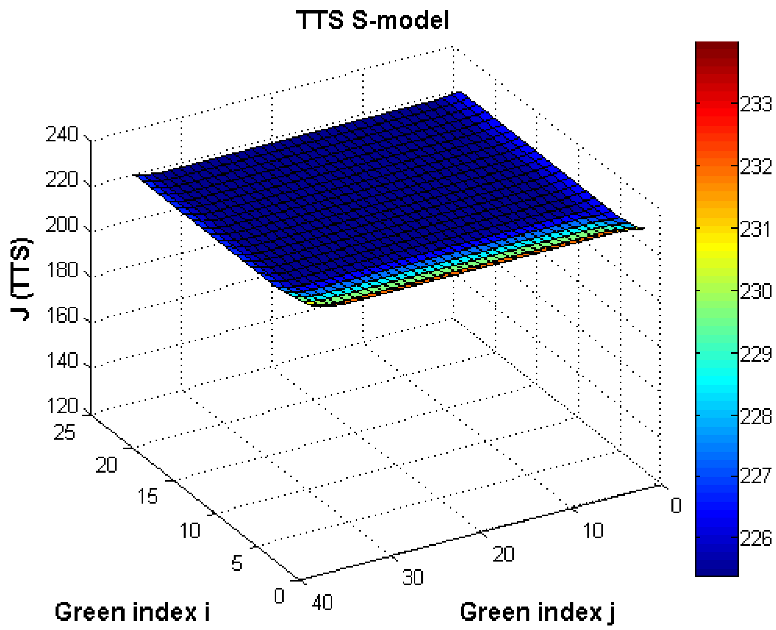


Figure 5.18 - Total Time Spent for the S-model, loaded with traffic pattern 2



In Vissim, a green time lower than 14s will cause queuing, this is according to what we expect. For the value i=18, the same can be done. This value of i could become critical for the directions 7, 8 and 9 (see Figure 5.16). Direction 10 is also part of the phase considered, but is present in another phase as well. Therefore direction 10 is not considered to be influenced by variable i. The flow present there is 275veh/h per lane, so the expected minimum green time would be: 275*(3600/1800)*(90/3600) = 13.75.
In Vissim, queuing starts at i=18, which correspond to a green time of 17s. The difference of 3s could be caused by the length of the left and right turning lanes. Their length is just sufficient to carry the average

number of arriving vehicles. Off course, the arrival rate is not distributed completely constant, so some blocking of other directions can occur. See Figure 5.7 for an example and Section 5.1.1 for the explanation.

For variable j, the expected lengths of green times can be calculated in a similar way. The corresponding green time to j=8 is 19s for directions 1 and 2 (direction 8 is also present in the fixed phase 3), see Figure 5.2. The expected value of the green time, (the flow is 290veh/h/lane) is:
290*(3600/1800)*(90/3600) = 14.5s.
For j=33 (and bigger), the corresponding green time is 16s (or smaller). The expected minimum would be (the flow is 250veh/h/lane):
250*(3600/1800)*(90/3600) = 12.5s. The difference between the theoretical and practical value can be explained due to blocking, see Figure 5.7. The explanation is similar to the one given for pattern 1.

When comparing the TTS from both Vissim and the S-model, one can see similar differences as with traffic pattern 1. The TTS is constant for values of i larger than 5, and for values of j between 3 and 34. The difference between the lowest and highest value for TTS is around 4%, which can be considered small. It is remarkable that the S-model predicts a change in TTS where i is small, but not when i has its highest value of 23. Vissim predicts the biggest change in TTS (with respect to variable i) in the region where i=23. The S-model predicts no change at this point. The reason must be that the amount of traffic flowing at directions 7, 8 and 9 (see Figure 5.3) is of such a low value that a green time of 12s is enough to process the traffic. The flow is thus lower than the demand, because already earlier it was shows that the minimum green time to process the traffic is 14s. This means systematically, queues are building up for every possible combination of i and j. The high value of TTS for the S-model is a direct consequence of this issue. When looking at the Vissim graph (Figure 5.17) on the other hand, the dark blue area where the value of TTS is lower than 130 can be considered congestion free (no queues are building up).

5.1.3 Traffic load pattern 3

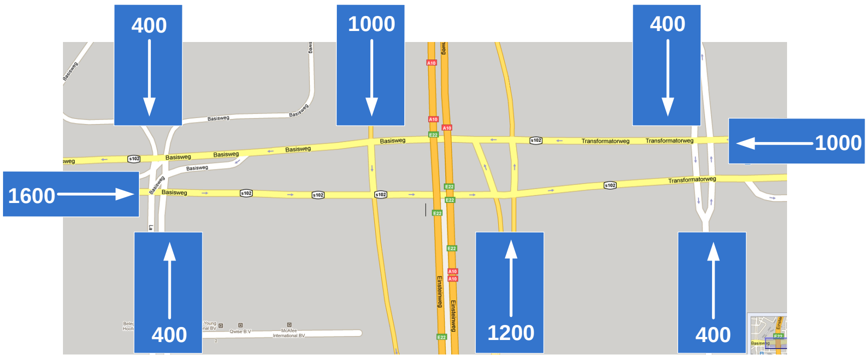


Figure 5.19 – Traffic load pattern 3. Flows are in vehicles/hour.

Figure 5.20 – Total Time Spent for the Vissim model, loaded with traffic pattern 3

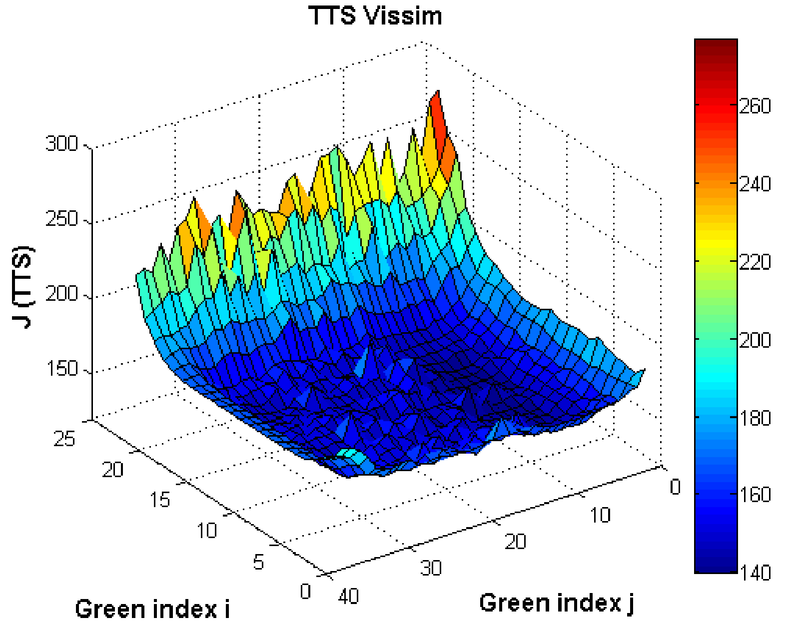
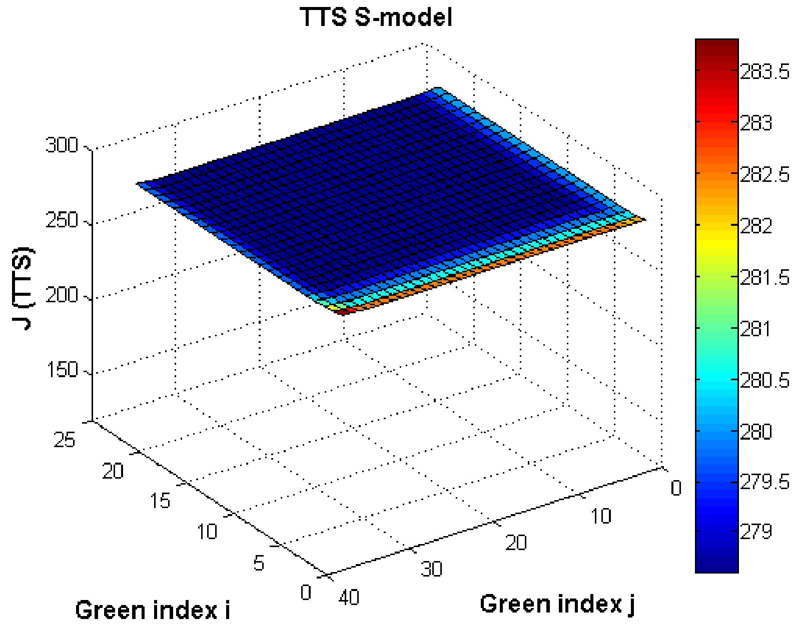


Figure 5.21 - Total Time Spent for the S-model, loaded with traffic pattern 3



The third traffic load pattern (Figure 4.24) is of the same kind as pattern 2 (busy traffic on the main roads, less busy traffic on the side roads). When comparing the TTS of Vissim (Figure 5.20) and of the S-model (Figure 5.21) to pattern 2, the figures are roughly the same. The main difference is that the values for TTS are higher. The patterns are similar. The highest value of TTS for the S-model is 2% higher than the lowest value, a small difference.

Just as in pattern 2, TTS in the S-model does not increase when variable i gets large. This is because only limited flow can occur at directions 7, 8 and 9 (Figure 5.3). Therefore, the S-model will also return a high value of TTS for more favourable values of variable i. For more explanation, see pattern 2.

5.1.4 Traffic load pattern 4

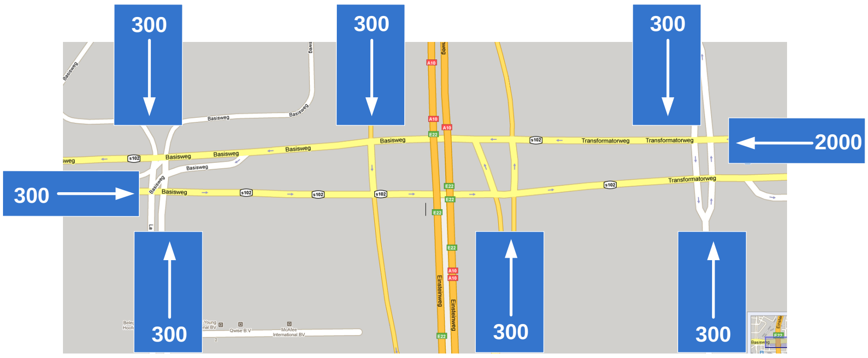


Figure 5.22 – Traffic load pattern 4. Flows are in vehicles/hour.

This pattern represents event traffic (Figure 4.25). The event traffic stream is heavy. The other are traffic streams are non-heavy to such a degree that their flows (300veh/h) will never create traffic jams, even when the green times have their lowest possible value (in this case 12s).

The biggest difference between Vissim (Figure 5.23) and the S-model (Figure 5.24) is the partly insensitivity to i. Between i=14 and i=23, the value of TTS is constant. For values of i lower than 14, the values of TTS are rising in the fashion as for Vissim. For this range in variable i, the S-model can be considered to give accurate predictions.

In Vissim the TTS is lowest when i is highest, this is an result that one would expect. When i is high, the only busy traffic stream gets the largest green time. The sensitivity to variable j is from j=10 (and smaller values). This corresponds to a green time of 21s. This is the minimum green time required to process the incoming traffic at intersection Kabelweg-Transformatorweg.

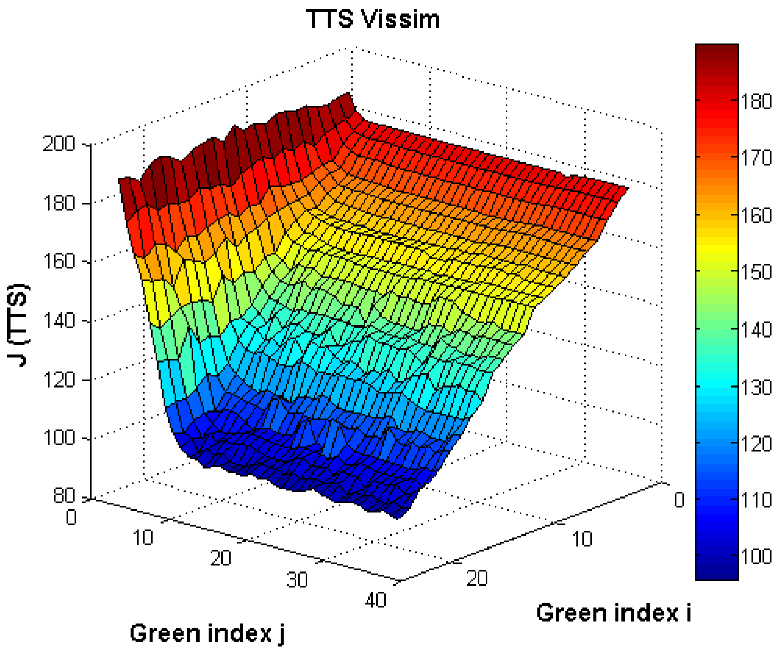


Figure 5.23 – Total Time Spent for the Vissim model, loaded with traffic pattern 4

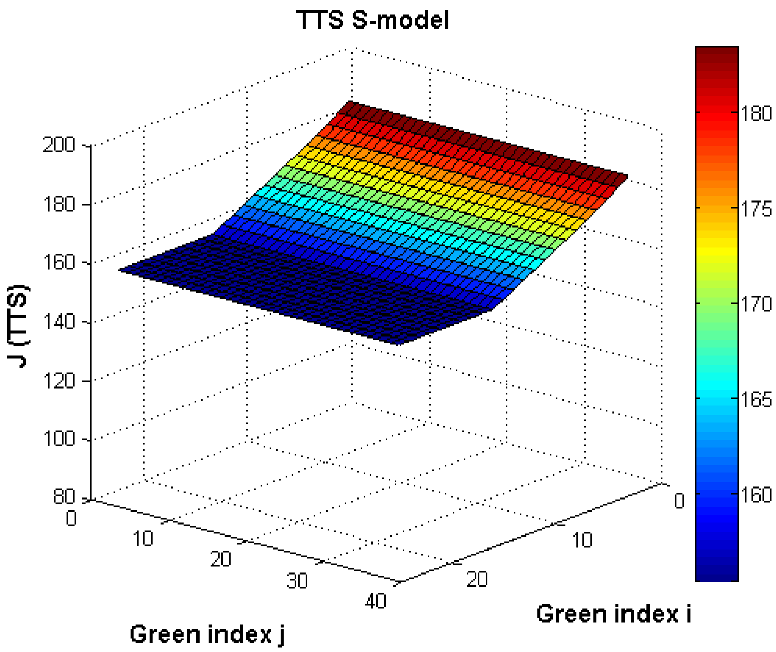


Figure 5.24 - Total Time Spent for the S-model, loaded with traffic pattern 4

Insensitivity by index j in S-model is caused by the fact that too few vehicles can leave the Transformatorweg and enter the link towards both the A10 and Basisweg (directions 1 and 2 in Figure 5.2). The Vissim graph clearly shows that a minimum in TTS can only be obtained if j is at least 10, which corresponds to a green time of 21 seconds. Because of this, it can be understood that because only limited flow can enter this link, a relatively short green time is sufficient to process the traffic.

5.1.5 Conclusion

In general it can be said that there is a difference between Vissim and the S-model with respect to calculated TTS. It appears that for this sensitivity analysis, a limiting factor in the S-model is the amount of traffic that is able to enter an upstream link. This causes queuing in situations where that would be not expected. For example in traffic pattern 1 (all entry flows are 600 veh/h), with average values for the green times (i=11, j=18), no congestion (queuing) should occur. In the S-model, queuing does occur at the links where traffic enters the network.

Because of the occurrence of queuing at relatively low entry flows (in the first traffic pattern with flows of 600veh/h, queuing is not expected), it can be seen in the TTS graphs for the S-model, that the lowest value for TTS is not a single value but a range of values. In the graphs, this is represented by a dark-blue coloured surface.

When considering the S-model to be used in an MPC controller, a control signal is optimized by minimizing the objective function; minimizing TTS. From this sensitivity analysis, it is clear that when TTS is minimized using the S-model (with all specific parameters used for the S102 network lay-out), there are many optimal solutions; solution in which TTS is lowest. From these optimal solutions, only a limited number of them is an optimal solution in Vissim. The accuracy of a prediction using the S-model is too low to be valid in Vissim. Therefore the S-model (in the current state) and Vissim are considered incompatible.

The consequence of this result is that in experiment 1, the proposed tests in which MPC is involved, cannot be run. Experiment 2, in which only MPC is used as a control method, cannot be run at all.

5.2 Experiment 1: Different control strategies compared

The results for this experiment will consist of the MFD and a table containing the following performance indicators for both the fixed time and Vrigen controlled networks:

- Total time spent in the network (h)
- Total distance travelled (km)
- Average speed (km/h)
- Total delay (h)
- Average delay time per vehicle (s)
- number of vehicles that left the network

Results for the MPC controller are not available due to incompatibility between the S-model and Vissim. Both a peak hour traffic pattern and an event traffic pattern were investigated.

For each combination of traffic load pattern and control strategy, five runs have been made. The value of the random seed in Vissim is chosen as their denominator, which ranges from 20 to 24. For all five runs, the MFD's are plotted in two similar graphs; the first graph is made with dots. The other graph is made using lines while also a trend line is added by hand. If the reader wishes to see individual graphs, they can be found in Appendix E.

5.2.1 Peak hour traffic pattern, fixed time control

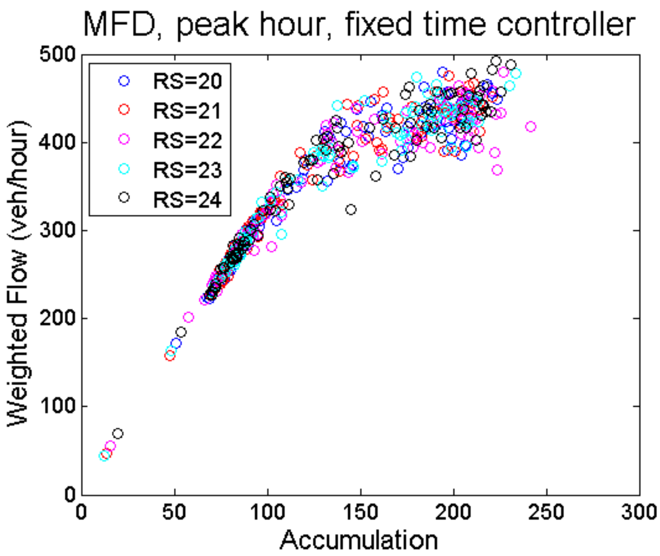


Figure 5.25 – MFD for peak hour traffic, fixed time traffic control. For each run, one MFD is plotted (5 in total) with dots.

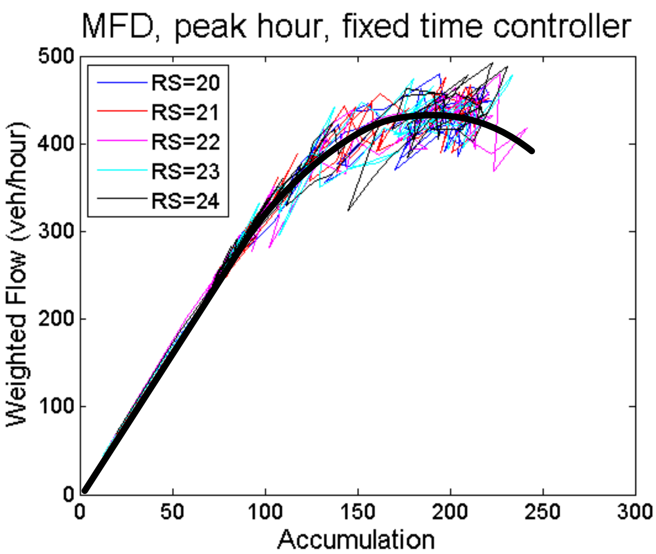


Figure 5.26 – MFD for peak hour traffic, fixed time traffic control. For each run, one MFD is plotted (5 in total) with lines. The black line is the trend line.

Table 5.2 – Performance indicators for fixed time control applied to peak hour traffic

Performance indicator	Rs=20	Rs=21	Rs=22	Rs=23	Rs=24	Average
Total time spent (h)	1246	1172	1255	1204	1183	1212
Total distance travelled (km)	31474	31490	31495	31466	31469	31479
Average speed (km/h)	25,3	26,9	25,1	26,1	26,6	26,0
Total delay time (h)	697	622	705	654	634	663
Average delay time per vehicle (s)	127	113	129	119	116	121
Number of vehicles that left the network	19494	19502	19504	19477	19459	19487

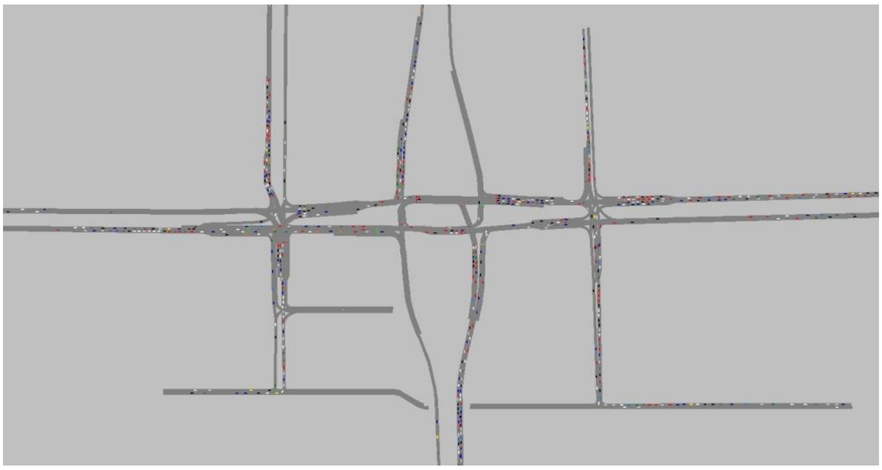


Figure 5.27 – Queues present at all entry links during peak hour traffic simulation (fixed time control)

From the MFD's in Figure 5.25 and Figure 5.26, a typical MFD shape can be seen. The five runs show a similar pattern: all MFD's are linear graphs until the point [100,300]. Then, there is a somewhat scattered area is seen where the saturation flow is located around the point [190,225]. At accumulation=200, the value for flow ranges between 390 and 450. There is no reference for the level of scattering in an MFD that determines whether an MFD can be considered constant. In Geroliminis and Daganzo (2008), the variation in their findings is around 40veh/h, while their time steps are much larger (values are smoothed more) than the cycle time of 90s used in this thesis. It is therefore argued that the MFD in constant for this fixed time controller. This result supports the findings by Geroliminis and Daganzo (2008) in which the existence of a constant MFD for an urban road network is argued.

Figure 5.27 shows a snapshot during the simulation in Vissim. It can be seen that long queues did form in all entry links. This is because of the traffic pattern; the peak in traffic demand cannot be processed in such a way that all vehicles waiting in front of a traffic light can leave the same cycle. Therefore, queues are growing. Because of these queues, delay is experienced by the vehicles. However, hardly any congestion due to blocking of downstream links did occur. Therefore a stable outflow could be reached, preventing congestion. This can be seen in the MFD's, because a higher accumulation did not cause the flow to drop sharply; there is no capacity drop found, which is usually found on regular fundamental diagrams of a road section.

The reason is that traffic tries to prevent blocking. In some countries, road users are encouraged or obliged no to block an intersection by indicating the conflict areas with a yellow cross, at which stopping is not allowed (see Figure 5.28).

Figure 5.28 – Conflict areas indicated by a yellow cross. Stopping in these areas is not allowed.

This principle is also applied in Vissim. In Figure 5.29 the conflict zone is indicated by a light green surface. The direction indicated in green has priority with respect to the direction in red. In Figure 5.30, one can see the consequence: no matter how long the queue, the conflict area will be kept clear so traffic in other directions is not blocked. This principle might seem unrealistic, because in reality, not everybody will keep the conflict area free. In Vissim, vehicles will randomly violate this obligation because each vehicle will individually assess whether an intersection can be crossed without causing a blocking. Most vehicles will make an accurate estimation while others do not. When the vehicle fails to make an accurate estimation, blocking will occur.

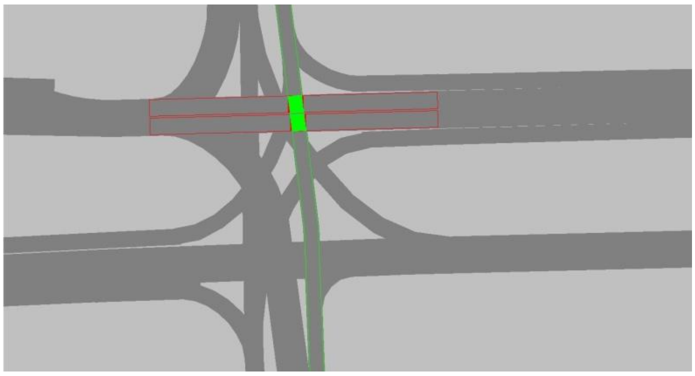


Figure 5.29 – Conflict area indicated by green and red surfaces in Vissim at intersection Kabelweg-Transformatorweg. The green direction has priority.

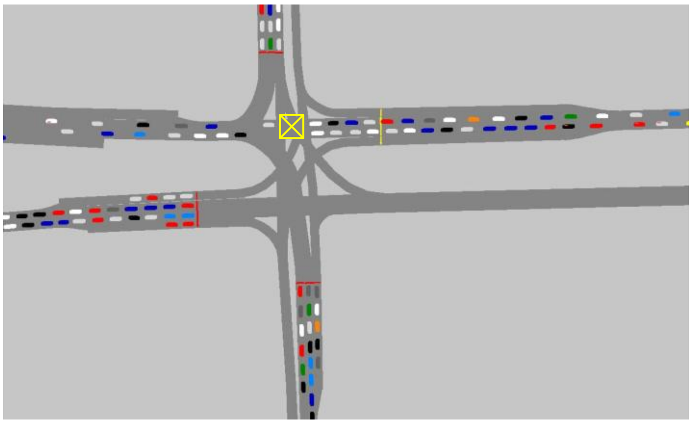


Figure 5.30 – Conflict area indicated in Figure 5.29 is respected by the road users that keep the yellow box clear.

Another factor that contributes to the prevention of blocking is the coordination of the traffic signals. The traffic control schemes are chosen in such a way that heavy traffic streams are coordinated. Let us take westbound traffic as an example (Figure 5.31). Traffic originating from the A10 and Transformatorweg in western direction, have both coordinated traffic signals. The only way in which these traffic flows will cause queues at one the Basisweg-A10 intersections, is a downstream blocking, for example at Basisweg-La Guardiaweg as indicated in Figure 5.31. Then, only limited outflow is possible there (indicated with red arrows). In the previous paragraph it was explained that blocking is prevented as much as possible, therefore queuing in the downstream intersections is not likely to happen.

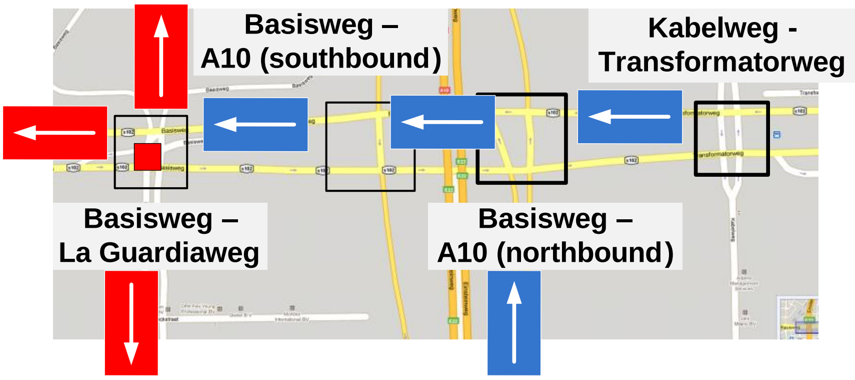


Figure 5.31 – A blocking at intersection Basisweg-La Guardiaweg influences the outflow of westbound traffic.

The question might rise if the blocking of the intersections is modelled in a realistic way. It is argued that the answer to this question be yes. When a higher degree of blocking is allowed, a gridlock is very like to happen. When a gridlock happens, the results in terms of performance indicators do not say anything. The only indicator of some use will be the jam density indicated on the MFD. So in order to get useful results, is it essential that situation prone to gridlock are avoided.

Because of the coordinated control and the prevention of blocking, a heavier traffic pattern is not expected to yield different MFD's. Since the control timers do not change, the queues will get larger, the delay time will increase and the average speed will decrease. This has been additionally tested on the traffic load pattern, but different scaling factors where the peak is more intense (see Figure 3.10).

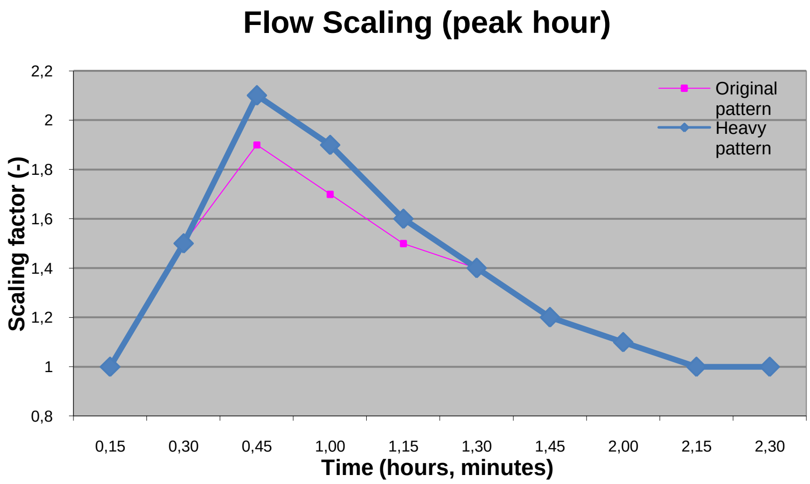


Figure 5.32 – Scaling of the flows in peak hour traffic

In Table 5.3, the performance indicators are present of this additional test. The last column indicates the difference in performance between the average values of the original pattern with respect to the heavy pattern. One can see that the heavier peak increases the total number of vehicles in the network by 4%, while the average delay time experienced rises by 43% (Table 5.5). Even so, the MFD's measured are quite similar (Figure 5.33 and Figure 5.34) with respect to the MFD's in Figure 5.25 and Figure 5.26. Both are a linear function up to the point [100, 300] and the maximum flow is scattered around [200, 425]. No significant drop of the flow due to increasing accumulation is indicated. The expectation formulated earlier ("a heavier traffic pattern is not expected to yield different MFD's", "the queues will get larger, the delay time will increase and the average speed will decrease") is proven to be correct.

Table 5.3 – Performance indicators for fixed time control applied to peak hour traffic (heavy pattern)

Performance indicator	Rs=20	Rs=21	Rs=22	Rs=23	Rs=24	Average
Total time spent (h)	1556	1527	1566	1556	1567	1555
Total distance travelled (km)	32659	32670	32698	32664	32654	32669
Average speed (km/h)	21,0	21,4	20,9	21,0	20,8	21,0
Total delay time (h)	986	957	996	986	998	985
Average delay time per vehicle (s)	173	168	175	173	175	173
Number of vehicles that left the network	20228	20244	20259	20241	20197	20234

Table 5.4 – Performance indicators for fixed time control applied to peak hour traffic compared

Performance indicator	Average original pattern	Average heavy pattern	Difference (%)
Total time spent (h)	1212	1555	28%
Total distance travelled (km)	31479	32669	4%
Average speed (km/h)	26,0	21,0	-19%
Total delay time (h)	663	985	49%
Average delay time per vehicle (s)	121	173	43%
Number of vehicles that left the network	19487	20234	4%

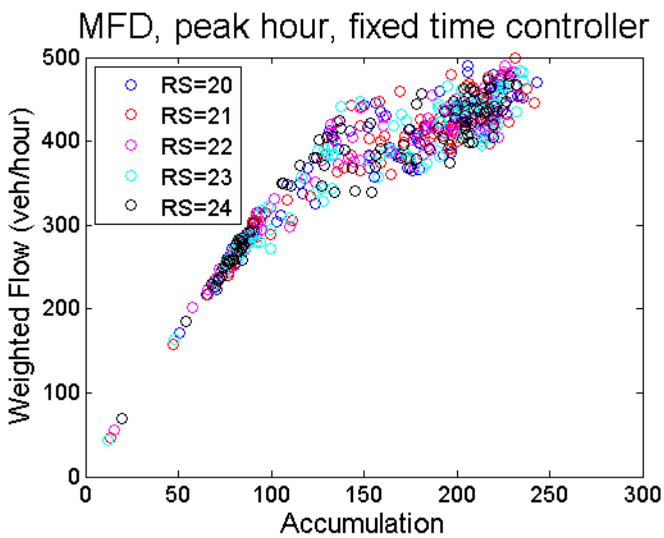


Figure 5.33 – MFD for peak hour traffic, fixed time traffic control. For each run, one MFD is plotted (5 in total) with dots.

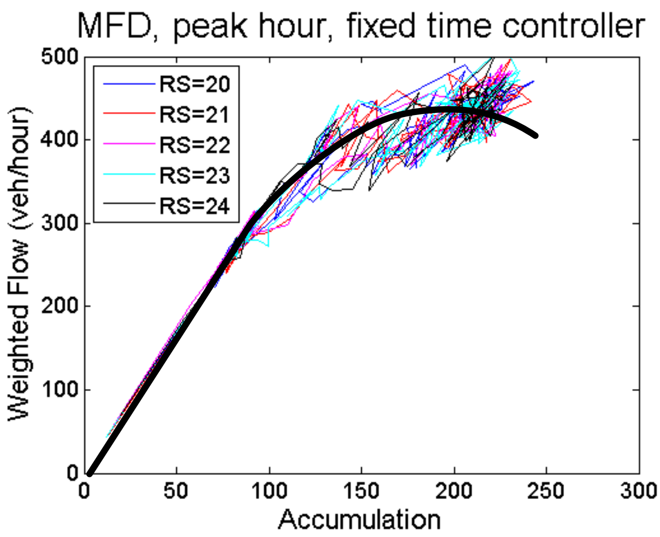


Figure 5.34 – MFD for peak hour traffic, fixed time traffic control. For each run, one MFD is plotted (5 in total) with lines.

5.2.2 Peak hour traffic pattern, Vrigen control

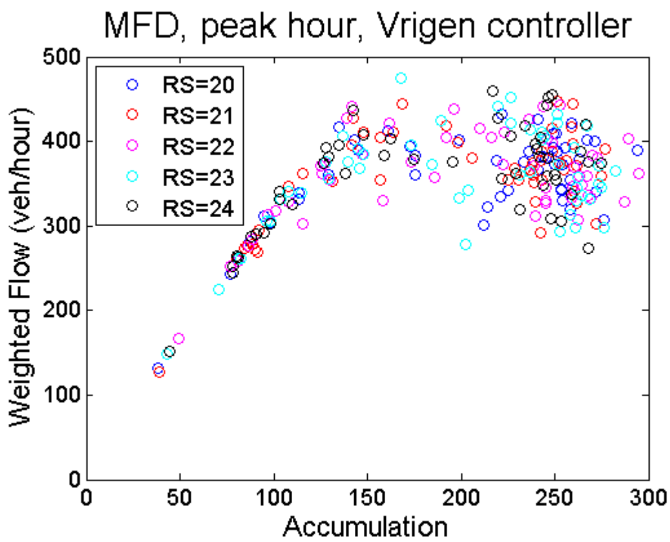


Figure 5.35 – MFD for peak hour traffic, Vrigen control. For each run, one MFD is plotted (5 in total) with dots.

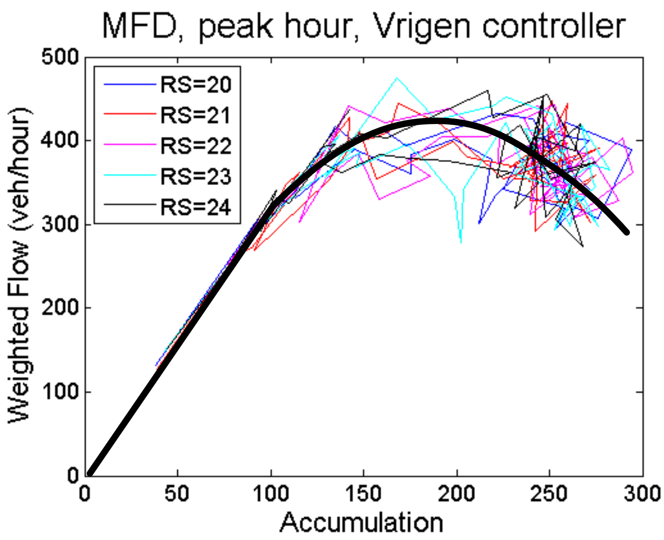
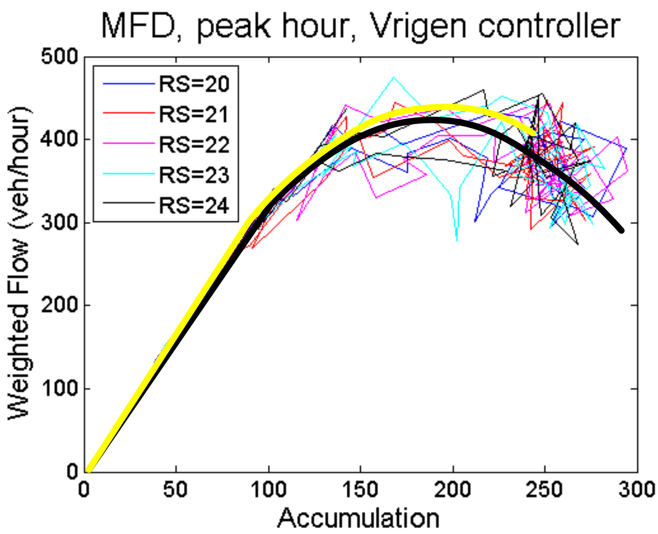


Figure 5.36 – MFD for peak hour traffic, Vrigen control. For each run, one MFD is plotted (5 in total) with lines.

Figure 5.37 – Trend lines from both Vrigen control (black) and fixed time control (yellow)



For the Vrigen controller applied at a peak hour traffic pattern, the MFD's can be found in Figure 5.35 and Figure 5.36. The free flow part (until accumulation is around 100) is followed by a scattered pattern of dots. At accumulation=250, the flow varies between 300veh/h and 450veh/h. The level of scattering is higher when compared to fixed time control. The global shape however is not so different (Figure 5.37). The MFD for fixed time control (yellow) has a higher saturation flow, while the MFD for Vrigen control shows flows for higher values for accumulation, indicating the congestion part of the MFD. For all five runs, it can be seen in the MFD's that from a state of congestion, recovering to free flow state, the MFD is slightly lower. There is a capacity drop noticeable. It is visible for example in Figure 5.36 for RS=23.

Table 5.5 – Performance indicators for Vrigen control applied to peak hour traffic

Performance indicator	Rs=20	Rs=21	Rs=22	Rs=23	Rs=24	Average
Total time spent (h)	2153	1921	2003	1863	1738	1936
Total distance travelled (km)	31394	31473	31455	31379	31476	31435
Average speed (km/h)	14,6	16,4	15,7	16,8	18,1	16,3
Total delay time (h)	1606	1373	1454	1315	1189	1387
Average delay time per vehicle (s)	293	250	265	240	217	253
Number of vehicles that left the network	19395	19510	19478	19413	19482	19456

Table 5.6 – Performance indicators for fixed time control (original loading pattern) and Vrigen controller compared

Performance indicator	Average fixed time controller	Average Vrigen control	Difference (%)
Total time spent (h)	1212	1936	59,7%
Total distance travelled (km)	31479	31435	-0,1%
Average speed (km/h)	26,0	16,3	-37,3%
Total delay time (h)	663	1387	109,2%
Average delay time per vehicle (s)	121	253	109,1%
Number of vehicles that left the network	19487	19456	-0,2%

While the shape of the MFD's look not so much different when compared to fixed time control, it could be noticed that many states measured were in the congested part of the MFD. The performance indicators show the consequence: the total distance travelled and the number of cars leaving the network are virtually equal. The average delay per vehicle however is more than doubled and the total time spent is increased by 60%.

This difference can be mostly explained by the fact that coordination is only present at the Basisweg-A10 ramp intersection. To the other two intersections, there is no coordination at all. The philosophy of Vrigen is that controlling optimally on intersection level makes coordination between intersections irrelevant. Because of the proximity of all intersections, queuing space between them is relatively limited. When heavy traffic is present, this space will be filled quickly and be overloaded if it can't be cleared in time. If this happens, there's blocking. This happens often when Vrigen control is applied to the peak hour traffic pattern.

To create coordination, the term "offset" is important. It is the shift in time between the cycles of adjacent intersections. In fixed time control, this offset can be predefined and tuned perfectly. In Vrigen control, offset values are random and not controllable. This is because the cycle time is only partially controllable and offsets cannot be defined. A maximum cycle time is defined, but it depend on the actually traffic detection whether this cycle time will be reached or will be shorter. The consequence is a lower average waiting time on a local (intersection) level, but also the chance to give a green signal at an unfavourable moment. It is well possible that green is such that it turns red when a busy traffic stream is just arriving which has to queue and will exceed the queuing space. This happens very frequently, see Figure 5.38 for some examples. It is this lack of queuing space that causes the MFD's to show congestion; the flow is limited because, as opposed to the fixed time control case, the outflow is not more or less free flow.

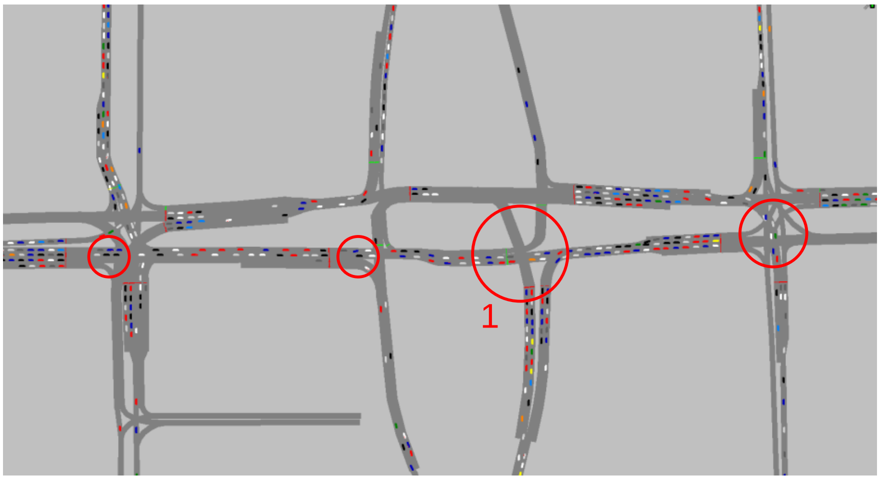


Figure 5.38 – Vehicles in the red circles are queuing at the intersections since the queuing spaces are full. The green light given in red circle 1 is useless.

Please be aware that this result for Vrigen is true for this specific traffic pattern. For less heavy peak hour traffic patterns, the difference in performance (with respect to fixed time control) will be smaller.

5.2.3 Event traffic pattern, fixed time control

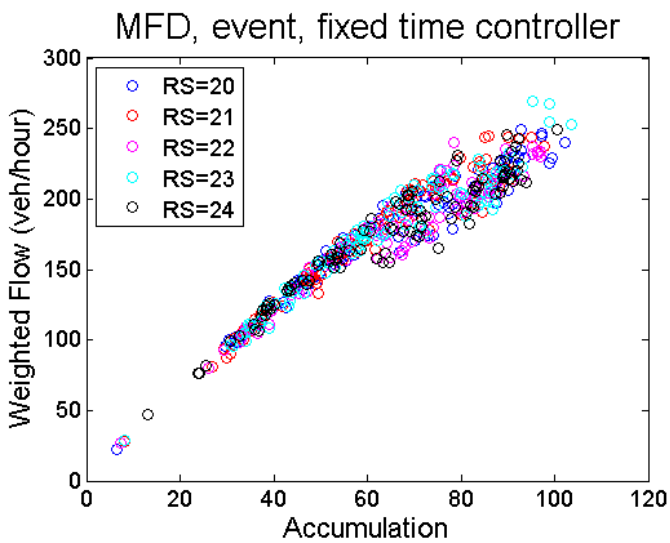
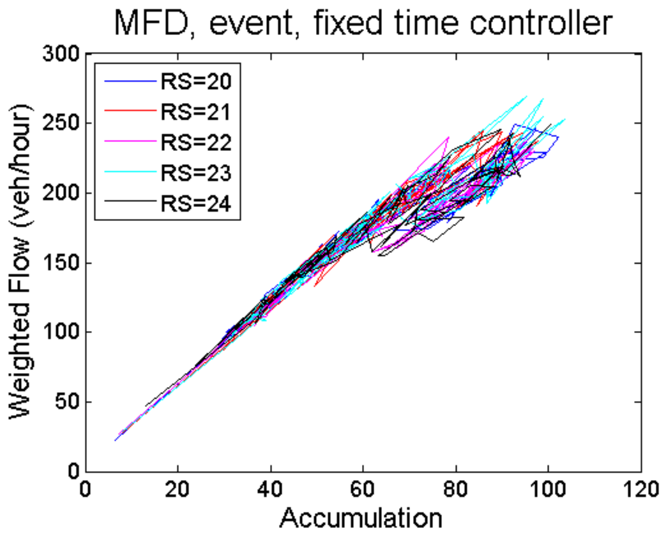


Figure 5.39 – MFD for event traffic, fixed time control. For each run, one MFD is plotted (5 in total) with dots.

Figure 5.40 – MFD for event traffic, fixed time control. For each run, one MFD is plotted (5 in total) with lines.



The MFD's in Figure 5.39 and Figure 5.40 show similar patterns for all five runs. Unlike in the peak hour traffic cases, no trend line is indicated as the variation is small. In this traffic pattern, congestion due to blocking is highly unlikely, because only two entry flows can be considered heavy while the others are light (300veh/h). An exception is a situation in which the signal control timers are poorly chosen. In that case, a queue blocking other traffic is possible.

The MFD's are different when compared to the same control strategy in peak hour conditions. For the timer settings in event traffic, the flow at accumulation=100 is around 250veh/h. In peak hour condition this is 300veh/h.

Table 5.7 – Performance indicators for fixed time control applied to event traffic

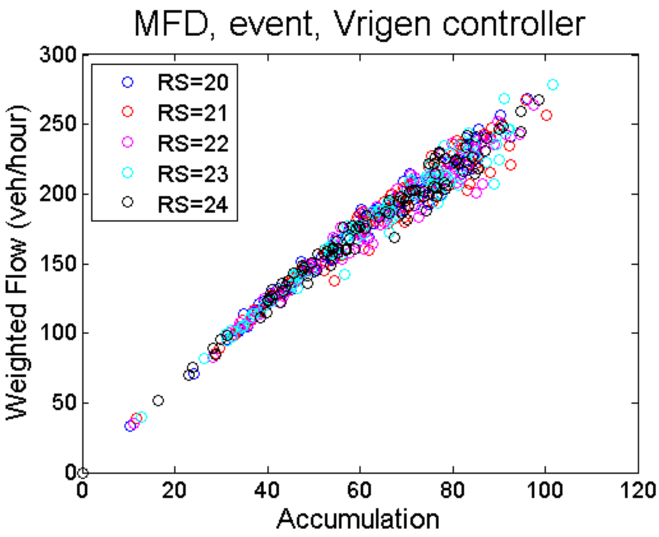
Performance indicator	Rs=20	Rs=21	Rs=22	Rs=23	Rs=24	Average
Total time spent (h)	547	529	555	539	546	543
Total distance travelled (km)	19733	19718	19733	19732	19674	19718
Average speed (km/h)	36,0	37,3	35,6	36,6	36,1	36,3
Total delay time (h)	180	162	188	172	180	176
Average delay time per vehicle (s)	59	53	61	56	59	58
Number of vehicles that left the network	10884	10872	10892	10881	10826	10871

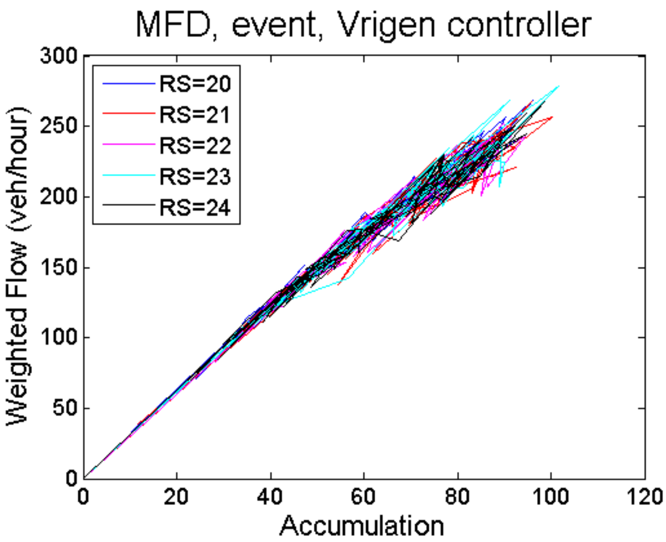
The performance indicators can be found in Table 5.7. These will be compared to the performance indicators in Vrigen control.

5.2.4 Event traffic pattern, Vrigen control

The MFD's in Vrigen are equal to those in fixed time control for the sections which are measured (free flow part of the MFD). For both fixed time and Vrigen control, no state of congestion was found, nor was a saturation state reached.

Figure 5.41 – MFD for event traffic, fixed time control. For each run, one MFD is plotted (5 in total) with dots.





This MFD (measured in an event traffic pattern) is different when compared to the Vrigen MFD in peak hour conditions (in the same way as for fixed time control); the MFD's in peak hour conditions are steeper. Their linear part of the MFD intersects with point [100,300] while the linear part of the event traffic MFD's intersect with point [100,250].

Table 5.8 – Performance indicators for Vrigen control applied to event traffic

Performance indicator	Rs=20	Rs=21	Rs=22	Rs=23	Rs=24	Average
Total time spent (h)	539	537	548	567	535	545
Total distance travelled (km)	19972	19955	19971	19980	19918	19959
Average speed (km/h)	37,0	37,2	36,4	35,3	37,2	36,6
Total delay time (h)	168	165	177	195	165	174
Average delay time per vehicle (s)	54	54	57	63	53	56
Number of vehicles that left the network	11005	11008	11026	11026	10968	11007

Table 5.8 shows the performance indicators for Vrigen control applied to event traffic. In Table 5.9, the difference between Vrigen en fixed time control is given. It is found that the differences are minimal. For Vrigen, a smaller delay time would be expected for the non-busy traffic streams. When traffic volumes are small, Vrigen will yield low cycle times (much lower than 90s in the fixed time case). On the other hand, coordination for the busy traffic streams is not present. Finally the difference is 3,4% in favour of Vrigen.

Table 5.9 – Performance indicators for fixed time and Vrigen control compared for event traffic

Performance indicator	Average Fixed time control	Average Vrigen control	Difference (%)
Total time spent (h)	543	545	0,4%
Total distance travelled (km)	19718	19959	1,2%
Average speed (km/h)	36,3	36,6	0,8%
Total delay time (h)	176	174	-1,1%
Average delay time per vehicle (s)	58	56	-3,4%
Number of vehicles that left the network	10871	11007	1,3%

5.2.5 Conclusion

For two differently scaled peak hour traffic patterns, the MFD is constant while the traffic performance is influenced significantly; a 4% increase in total number of vehicles increases TTS by 28%. No state of congestion is reached, since only queuing occurs, but no blocking. This is due to coordination of the traffic signals as well as vehicles that keep conflict areas clear.

When applying Vrigen control to the peak hour traffic pattern, the shape of the MFD is could be considered constant. However, after a state of congestion the MFD will show a slight capacity drop. Congestion does occur when applying Vrigen control, because no coordination is present. This can cause queuing inside the network because the intersections in the S102 network are close to one another (<150m). When considering fixed time control, offsets can be given while for Vrigen control offsets are random. The difference in network performance is big: Vrigen increases TTS by 60%, while the amount of vehicles that left the network is more or less equal with respect to fixed time control.

For the event traffic pattern, both fixed time control and Vrigen control show equal network performance and equal MFD shapes. A state of congestion is unlikely in both cases, but this strongly depends on the exact flows in the networks and the network lay-out. When for example the distance between intersections is shorter, the relative importance of coordination will increase because less space is available for queuing.

6. Conclusions and recommendations

6.1 Introduction

At the start of this thesis, the following research questions were formulated:

Main research questions

1. How to design a controller that controls DTM measures on a subnetwork level in a hierarchical setting?
2. What is the difference in performance between coordinated control and conventional control methods?
3. Is it possible to use MFD in a control concept for subnetworks?

The three questions will be answered one by one.

6.2 Controller design in a hierarchical concept

This section will answer research question 1. It was explained that MPC is chosen as the control method to control a subnetwork in a hierarchical setting. The MPC controller consists of three parts: a prediction model, cost function(s) and an optimizer. The optimizer was selected to be fmincon (Matlab) because creating an optimizer is a specific task, which is beyond the scope of this thesis.

6.2.1 Cost function

A general cost function was formulated:

$$\min(J) \qquad \text{Eq. 6.1}$$

$$J = \alpha_1 J_{TTS} + \alpha_2 J_{Qref,out} + \alpha_3 J_{Qref,in} + \alpha_4 J_{MFD} \qquad \text{Eq. 6.2}$$

In which J_{TTS} is the penalty for total time spent, $J_{Qref,out}$ is the penalty for not obeying to desired flows going out of the network, $J_{Qref,in}$ is the penalty for not obeying to desired flows entering the network. J_{MFD} is the penalty for having a different shaped MFD with respect to the shape of the MFD used by the upper level controller.

6.2.2 S-model (prediction model)

The traffic prediction model selected in this thesis is the S-model. An important contribution of this thesis is that the S-model has been adjusted to function in a Windows environment together with programming software Matlab and microscopic traffic simulator Vissim.

The result is a technically working simulation environment in which Matlab, Vissim, the S-model and Trafcod work simultaneously together. Trafcod is a necessary link between Matlab and Vissim. It is a messenger to communicate the calculated control signal from the S-model by Matlab, to the traffic signal control in Vissim.

6.2.3 Sensitivity analysis between S-model and Vissim

After the simulation environment was technically working, a sensitivity analysis was carried out between the S-model and Vissim. The objective is TTS, the variables are green times. Four different traffic patterns were tested, each for 851 different combinations of green times. The sensitivity analysis showed that the two traffic models are not compatible at this moment. The reasons are described in detail in Section 5.1. When this issue was found out, time was too short to fix this problem. Therefore, the MPC controller cannot be used to conduct experiments.

6.3 Coordinated versus conventional control

Since the coordinated control cannot be used, the comparison of traffic performance between different control methods is limited to fixed time control and Vrigen control on two different traffic loading patterns. These traffic loading patterns are peak hour traffic and event traffic.

For two differently scaled peak hour traffic patterns, the MFD is constant while the traffic performance is influenced significantly; a 4% increase in total number of vehicles increases TTS by 28%. No state of congestion is reached, since only queuing occurs, but no blocking. This is due to coordination of the traffic signals as well as vehicles that keep conflict areas clear.

When applying Vrigen control to the peak hour traffic pattern, the shape of the MFD is could be considered constant. However, after a state of congestion the MFD will show a slight capacity drop. Congestion does occur when applying Vrigen control, because no coordination is present. This can cause queuing inside the network because the intersections in the S102 network are close to one another (<150m). When considering fixed time control, offsets can be given while for Vrigen control offsets are random. The difference in network performance is big: Vrigen increases TTS by 60%, while the amount of vehicles that left the network is more or less equal with respect to fixed time control.

For the event traffic pattern, both fixed time control and Vrigen control show equal network performance and equal MFD shapes. A state of congestion is unlikely in both cases, but this strongly depends on the exact flows in the networks and the network lay-out. When for example the distance between intersections is shorter, the relative importance of coordination will increase because less space is available for queuing.

6.4 MFD as a control concept in hierarchical control

The third research question is whether the MFD can be used as a control concept in a hierarchical control concept. Since no experiments have been done using MPC, this question cannot be answered. However, some theoretical outcomes will be discussed after the results from the conventional control techniques have been evaluated.

In the hierarchical control concept, the upper level controller should be as simple as possible. Therefore, it is necessary that the shape of the MFD is either constant, or that its dynamics are known; it should be known how the shape of the MFD changes when control actions are taken. The latter option will be computationally more expensive.

The results from the comparison between fixed time control and Vrigen control suggest that the shape of the MFD is constant. This is only based on the results in peak hour traffic. The MFD's in event traffic only showed information about the free flow part of the MFD, so the behaviour in congested conditions are not known.

When considering the outcome of the experiment for the MPC controller, two outcomes are possible: MFD will be constant or it will be not constant.

In case the MFD is constant, it can be used by the upper level controller to communicate commands to a subnetwork. The upper level controller will for example send a desired traffic state to a subnetwork, as well as information like flows at the boundary of the subnetwork.

In case the MFD is not constant, there are two options. Either the behaviour of the changing MFD is such that it can be accurately predicted by the upper level control, or not. When the change in shape from the MFD is predictable, the upper level controller can calculated its signal while taking into account the changing shape of the MFD. If the changing of shape is not predictable, then the subnetwork MPC controller should have a penalty in its objective function for obeying to a reference MFD shape. The prediction model must be capable of providing the right output to construct an MFD.

In all cases, the MPC controller will make a trade-off between different terms of the objective function. In a hierarchical setting, the objectives by the upper level control are most important. Local objective like minimizing TTS are less important.

6.5 Recommendations

This chapter will discuss and give recommendations about the S-model (6.5.1), Vissim (6.5.2) and the Vrigen/Trafcod combination (6.5.3). This report will end with recommendations and discussion on hierarchical control.

6.5.1 S-model

In this section recommendations will be given with respect to the S-model.

Link flow

The problems related to link flow are extensively described in Section 5.1.1. In short, the time step in the model should be such that it is shorter than the free flow travel time on the shortest link. Further, the way free space is assigned in a downstream link should be done different.

The time step is equal to one cycle time, which corresponds to a minimum link length of 1km if one assumes a cycle time of 90s and a free flow speed of 40km/h. Increasing the link length is not an option as this would make the model not suitable for typical Dutch road network lay-outs. Also, the model should fit the traffic situation and not vice versa. Decreasing the time step is a possibility, but this will increase computational time. A solution could be to introduce the possibility for a vehicle to cross more than one link during one time step (one cycle time). This will increase computational power, but it could be less time consuming when compared to the use of a lower time step.

At this moment, the saturation flow at a stop line and the turning rates are connected. This means the traffic load pattern should be adapted to this property of the S-model. For academic purposes, this is not a problem. When applying the model to a real traffic situation, it could be a problem since the chance is small that the traffic pattern matches the turning rates at each intersection. It is recommended to decouple this relationship. This will increase computational power but introduce necessary flexibility.

The recommendations listed in this section are essential to create a new version of the S-model program that is compatible with Vissim.

Start/stop model

The S-model assumes instant acceleration to the maximum speed and also instant moving vehicles. This implies that coordination will not be calculated properly. In reality, it can make quite a difference if two downstream traffic lights are coordinated in such a way that both will give a green signal. In the S-model, the traffic will have the same travel time, whether the vehicles have to stop or experience a “green wave”. Please note that the S-model will include blocking of the intersection.

At the start of this thesis, it was expected that “green waves” would appear naturally when heavy traffic streams are present when minimizing to TTS. This is not the case, because of the nature of driving behaviour in the S-model. It is therefore advised to include shockwave theory in order to model the effect of delay when vehicles want to drive away from a queue.

When implementing this feature, computational time will increase, so one will have to test the relation between accuracy and computational time and make a decision.

Clearance times

All clearance times are set to a standard value of two seconds for each conflict. In reality, this can yield over- or underestimating of an intersection’s capacity. It is possible that a decrease of 1 second in clearance time can mean a saving of 5 second on the cycle time. When an intersection is near capacity, this is a big difference. It is therefore advised to include variable clearance times to increase the realistic nature of the S-model. Implementing this feature will not increase the computational time of the S-model program.

Initial solution for optimization

When optimizing the S-model with fmincon, the value of the minimized objective function is highly dependent on the initial solution. Therefore multiple initial solutions should be tried.
For many possible sets of green times, the S-model will yield similar values for the TTS, see for example Figure 6.1. Then, it can be difficult for fmincon to search an optimum. It would be worthwhile to

investigate on other optimizers that have no (or less) problems with functions in which the derivatives are 0 for a large domain.

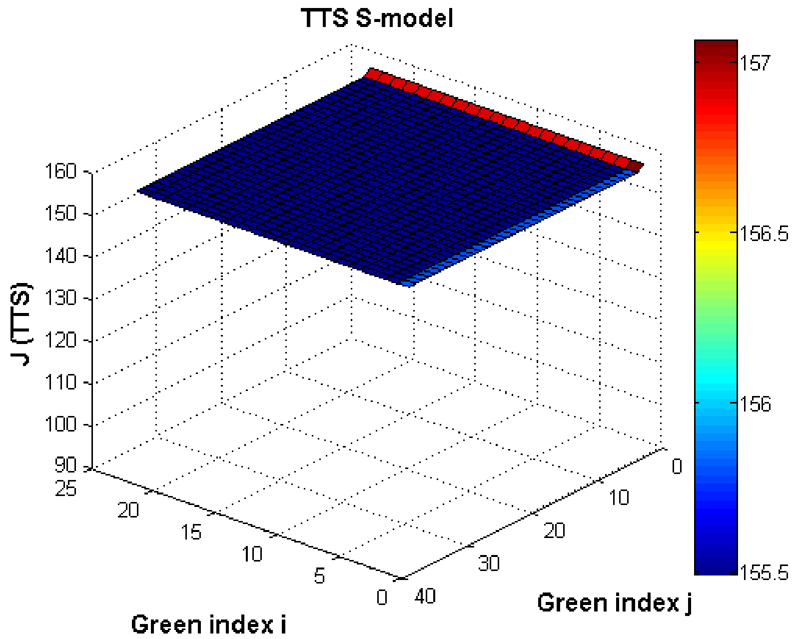


Figure 6.1 - Total Time Spent for the S-model

Phase sequence

The S-model works with time steps of one cycle time. At this moment offsets are not possible in the S-model. The term offset is defined as a time shift of the cycle time of one intersection with respect to another. This is used for the fixed time control and allows green wave patterns from traffic signal quite easily if all intersections have the same cycle times. Without offsets, there is limited control freedom to create coordination.

Since all phases start at the beginning of a cycle time, in principal all traffic signals show red. When using Trafcod, an exception is signal groups that do not belong to the critical conflict group (see section 6.5.3). This is a serious limitation in the current S-model. The effect can be minimized by choosing the phase sequence in such a way, that the most busy traffic streams at least get the opportunity of creating a green wave. This limitation is responsible for the difference in performance when compared to the fixed time control. If this limitation is eliminated, it is likely that the MPC controller will perform better than the fixed time control.

6.5.2 Vissim

The issue of intersection blocking is important. In reality, this can cause a limited outflow from other (the blocked) directions. In Vissim, conflicts have to be specifically indicated. This can be done with one of the two following functions:

- Conflict areas (Figure 6.2)
- Priority rules (Figure 6.3)

Conflict areas allow the user to indicate the priority direction. In theory, "They are the recommended solution in most cases because they are more easily defined and the resulting vehicle behavior is more intelligent." (PTV, 2009). The reality is that gridlocks become rule rather than the exception.

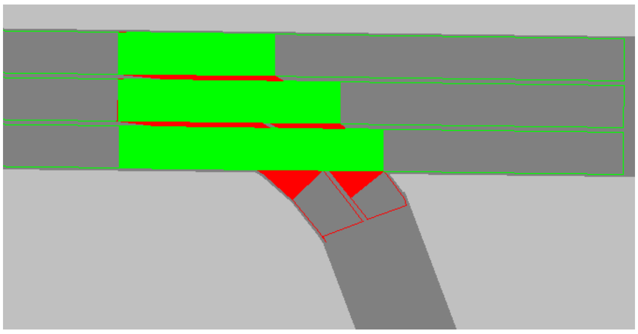


Figure 6.2 – Conflict areas in Vissim

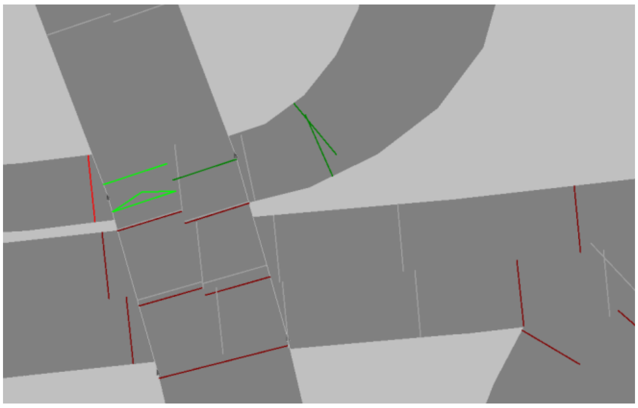


Figure 6.3 – Priority rules in Vissim

The author has tried to overcome the issue of unnecessary gridlocks with the function "priority rules", see Figure 6.3. The (relatively small) green area is the conflict area. When a vehicle is present with a speed lower than 13km/h, the vehicle from the conflicting stream will wait in front of the red stop line. The conflict are is chosen small, so in case a car has just passed the green conflict area, it is still in the way but the vehicles in the conflicting stream can pass. This strategy seems to work up to a certain level. Still, gridlock situations are likely to occur in congested traffic. This is not realistic, since in reality this gridlock will not stay there forever. It would be worthwhile to have better priority rules in order to prevent gridlocks.

6.5.3 Vrigen/Trafcod

The usage of Trafcod as an external traffic controller for Vissim should be considered. During this thesis, there was no practical alternative available. However, it would be useful to switch to another external controller.

Trafcod and Vrigen are designed to control an intersection autonomously, not to simply execute commands from above. Problems arise for example when a signal groups has multiple realisations. There's only one value for a green timer, so there is already a problem. In this thesis, this is solved by setting the green timer of such signal groups to the value of guaranteed green, 4 seconds. Then, the option "parallel green" is selected. Parallel green means that a green signal is given as long a conflicting signal groups turn green.

When considering a network point of view, it could be desired that a signal group does not turn green, have only one realization, etcetera. From a network control point of view, one would like to have the possibility to have absolute control about the traffic signals. Trafcod was not designed to do so and will not provide that kind of control. Therefore, Trafcod is not the ideal external controller. The use of Vrigen is valuable in any case for creating phase schemes.

6.5.4 Hierarchical control

The use of MPC as a control concept in hierarchical traffic control is believed to have a high potential by the author. The quality of the signal depends most on:

- Available computational power
- Quality of the prediction model

Computational power will increase exponentially as the number of variables increases. The current S-model is only in the starting stage and has potential to predict traffic in a more accurate way.

The ideal view by the author would be a prediction model with the local intelligence of Vrigen while having a network approach. After all, this thesis only considered motorized traffic while bikers and pedestrians do influence the way intersections are controlled.

The use of MFD in hierarchical control is not clear yet. Evidence collected in this thesis is not enough to determine in a convincing way that the MFD has indeed a constant shape when the control method does not change. In order to be useful as a communication tool between upper and lower level controllers, the MFD should be a constant. Another, less favourable, option is that the dynamics of the MFD are known so the right MFD can be selected in the right circumstances. A changing MFD is less favourable because this involves extra computational effort and the possibility for errors when estimating the state. The next step in research to MPC based hierarchical traffic control could be to formulate the objective function of an MPC-controller in such a way that the shape of the MFD will stay constant.

In this thesis, the traffic state could be extracted from Vissim instantly within reasonable error margin (for most variables). In reality, state estimation has to be performed which leaves room for error. Further research on this topic will take time, so the author would hope for a field test for MPC based traffic control in 5 years from now.

Bibliography

Daganzo, C.F. and Geroliminis N., 2008. An analytical approximation for the macroscopic fundamental diagram of urban traffic. Transportation Research Part B 42, 771-781

Findeisen, R. and Allgöwer, F., 2002. An Introduction to Nonlinear Model Predicitve Control. 21st Benelux Meeting on Systems and Control, Veldhoven.

Geroliminis, N. and Daganzo, C. F., 2008. Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. Transportation Research Part B 42, 759-770

Godfrey, J.W., 1969. The mechanism of a road network. Traffic Engineering and Control 11 (7), 323-327

Hoogendoorn, S.P., 2007. Traffic Flow Theory and Simulation. CT4821 Lecture Notes. Delft, University of Technology

Lin, S., Schutter, B. de, Xi, Y., Hellendoorn, J., 2009. A simplified macroscopic urban traffic network model for model-based predictive control. Proceedings of the 12th IFAC Symposium on Transportation Systems, Redondo Beach, California, 286-291

Muller, T.H.J., 2004. Determining Sample Size, CT4822 course material, Delft University of Technology

PTV Vision, Vissim 5.20 User Manual

Rijskwaterstaat, 2009. Eindrapport PPA definitief, Dienst Verkeer en Scheepvaart

Venkat, A.N., 2006. Distributed Model Predictive Control: Theory and Applications. Ph.D Thesis, University of Wisconsin-Madison

Zuylen, H.J. van, Muller, T.H.J., Hegyi, A., Salomons, M., 2009. Dynamic Traffic Management, Traffic Control. Lecture notes CT4822, Delft University of Technology

Appendix A

Paper S-model

A simplified macroscopic urban traffic network model for model-based predictive control

S. Lin * **B. De Schutter** ** **Y. Xi** * **J. Hellendoorn** **

* *Department of Automation, Shanghai Jiao Tong University*
No. 800 Dongchuan Road, Minhang District, Shanghai, P. R. China
(e-mail: lisashulin@gmail.com, ygxi@sjtu.edu.cn).
** *Delft Center for Systems and Control, Delft University of Technology*
Mekelweg 2, 2628 CD Delft, The Netherlands
(email: b.deschutter@dcsc.tudelft.nl, j.hellendoorn@tudelft.nl).

Abstract: A model predictive control (MPC) approach offers several advantages for controlling and coordinating urban traffic networks. To apply MPC in large urban traffic networks, a fast model that has a low on-line computational complexity is needed. In this paper, a simplified macroscopic urban traffic network model is proposed and tested. Compared with a previous model, the model reduces the computing time by enlarging its updating time intervals, and preserves the computational accuracy as much as possible. Simulation results show that the simplified model reduces the computing time significantly, compared with the previous model that provided a good trade-off between accuracy and computational complexity. We also illustrate that the simplifications introduced in the simplified model have a limited impact on the accuracy of the simulation results. As a consequence, the simplified model can be used as prediction model for MPC for larger urban traffic network.

Keywords: Macroscopic traffic modeling; Urban traffic control; Model predictive control; Urban traffic network.

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. To reduce traffic jams and to promote efficiency in traveling, effective traffic control algorithms are necessary. Many control theories have been applied to control traffic (Kachroo and Özbay, 1999; Papageorgiou, 1983), like fuzzy control, PID control, model predictive control, and multi-agent control, in combination with different control structures like centralized, distributed, and hierarchical control.

We focus on model-based control methods, and on MPC in particular. Considering on-line computational complexity, macroscopic traffic models are usually used in MPC. However, for different model-based control approaches, there still exist different levels of requirements for the macroscopic model. Some models just need to express the relation between the input values and the performance indicators, but some are more detailed so as to describe the dynamics of the traffic evolution; some models are more precise in modeling the dynamics, while some are simpler so as to be fast for on-line computing. As a result, there exists a wide variety of macroscopic traffic models with different levels of detail. For different control methods, appropriate traffic models with the required modeling power need to be selected.

In the past few years, various macroscopic urban traffic models were developed and used for traffic control. The store-and-forward model, proposed by Gazis and Potts (1963) and later used by Diakaki et al. (2002), is a simple model with low computational complexity, but it can only be used for saturated traffic, i.e., if the vehicle queues resulting from the red phase cannot be dissolved completely at the end of the following green phase. The model proposed by Barisone et al. (2002) and extended by Dotoli et al. (2006) is computationally more intensive and it can describe different scenarios, but it is also more complicated. The model proposed by Kashani and Saridis (1983) has lower modeling power, but can not depict scenarios other than saturated traffic either. The model of van den Berg et al. (2003); Hegyi (2004); van den Berg et al. (2004) 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 compared with the microscopic model, which is tested and further extended in Lin and Xi (2008) and Lin et al. (2008).

In principle, a centralized MPC method guarantees globally optimal control actions for traffic networks. 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 significantly, when the network scale gets larger, the prediction time span gets longer, and the traffic model becomes more complex or gets a higher modeling power. There are two main approaches to address this problem: (1) simplifying the traffic model in order to reduce the computational complexity, and (2) cutting the traffic network into small sub-networks or even intersections, which are then controlled using distributed or multi-agent control. In this paper we consider

* S. Lin is a visiting researcher at the Delft Center for Systems and Control, Delft University of Technology.
**B. De Schutter is also with the Marine & Transport Technology department of Delft University of Technology.

the first approach, i.e., we develop simplified, yet sufficiently accurate, traffic models, in particular, for urban traffic networks.

We start with an urban traffic model based on previous work of Kashani and Saridis (1983); van den Berg et al. (2003); Lin and Xi (2008). To reduce the computational burden, the simplified model enlarges the simulation time interval to one cycle time. During each simulation time interval, traffic states are updated once in each link according to the average input and output traffic flow rates in the current cycle. To add flexibility, every intersection in the network can have a different cycle time, and the intersections share the same control time interval. This control time interval is the least common multiple of all the cycle times of the intersections in the network. It is necessary to define this common control time interval to keep the model running and communicating synchronously under both centralized control and distributed control. For a given link the average input traffic flow rates are provided by the upstream links, which transform their own output traffic flow rates into input flow rates for the given link taking the different simulation time intervals into account.

We will demonstrate with examples that this simplified traffic model reduces the simulation time significantly, compared with the model in van den Berg et al. (2003) and Lin and Xi (2008), with only a limited reduction in accuracy. This makes it possible to apply centralized MPC to larger urban traffic networks.

2. TWO MACROSCOPIC URBAN TRAFFIC NETWORK MODELS

In this section we present the original model of van den Berg et al. (2003) and Lin and Xi (2008) (indicated as the BLX model) as well as a new simplified model (called the S model). But first we introduce some common notation for both models.

Define J the set of nodes (intersections), and L 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 input and output links for link (u,d) are $I_{u,d} \subset L$ and $O_{u,d} \subset L$ (e.g., for the situation of Fig. 1 we have $I_{u,d} = \{i_1, i_2, i_3\}$ and $O_{u,d} = \{o_1, o_2, o_3\}$).

In order to describe the evolution of the models, we first define some variables (see also Fig. 1):

- $I_{u,d}$: set of input links of link (u,d),
- $O_{u,d}$: set of output links of link (u,d),
- k : simulation step counter for the urban traffic model,
- $n_{u,d}(k)$: number of vehicles in link (u,d) at step k,
- $q_{u,d}(k)$: queue length at step k in link (u,d), q_{u,d,o_m} is the queue length of the sub-stream turning to link o_m,
- $m^{\mathrm{l}}_{u,d,o_m}(k)$: number of cars leaving link (u,d) and turning to o_m,
- $m^{\mathrm{a}}_{u,d}(k)$: number of cars arriving *at the (end of the) queue* in link (u,d) at step k, $m^{\mathrm{a}}_{u,d,o_m}(k)$ is the number of arriving cars in the sub-stream towards o_m,
- $S_{u,d}(k)$: available storage space of link (u,d) at step k expressed in number of vehicles,
- $\alpha^{\mathrm{l}}_{u,d}(k)$: flow rate leaving link (u,d) at step k, $\alpha^{\mathrm{l}}_{u,d,o_m}(k)$ is the leaving flow rate of the sub-stream towards o_m,
- $\alpha^{\mathrm{a}}_{u,d}(k)$: flow rate arriving at the end of the queue in link (u,d) at step k, $\alpha^{\mathrm{a}}_{u,d,o_m}(k)$ is the arriving flow rate of the sub-stream towards o_m,
- $\alpha^{\mathrm{e}}_{u,d}(k)$: flow rate entering link (u,d) at step k,
- $\beta_{u,d,o_m}(k)$: relative fraction of the traffic turning to o_m at step k,
- $\mu_{u,d}$: saturated flow rate leaving link (u,d),
- $g_{u,d,o_m}(k)$: green time length during step k for the traffic stream towards o_m in link (u,d),
- $b_{u,d,o_m}(k)$: boolean value indicating whether the traffic signal at intersection d for the traffic stream in link (u,d) turning to o_m is green (1) or red (0) at step k,
- $v^{\mathrm{free}}_{u,d}$: free-flow vehicle speed in link (u,d),
- $C_{u,d}$: capacity of link (u,d) expressed in number of vehicles,
- $N^{\mathrm{lane}}_{u,d}$: number of lanes in link (u,d),
- $\Delta c_{u,d}$: offset between node u and node d,
- l_{veh} : average vehicle length.

2.1 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 $[kT_{\mathrm{s}}, (k+1)T_{\mathrm{s}})$ 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. 1). For each $o_m \in O_{u,d}$ the number of cars leaving link (u,d) for destination o_m in the period $[kT_{\mathrm{s}}, (k+1)T_{\mathrm{s}})$ is given by

$$m^{\mathrm{l}}_{u,d,o_m}(k) = \begin{cases} 0 & \text{if } b_{u,d,o_m}(k) = 0 \\ \max\big(0, \min(q_{u,d,o_m}(k) + m^{\mathrm{a}}_{u,d,o_m}(k), \\ \quad S_{o_m}(k), \beta_{u,d,o_m}(k)\cdot\mu_{u,d}\cdot T_{\mathrm{s}})\big) & \text{if } b_{u,d,o_m}(k) = 1\,. \end{cases}$$

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

$$m^{\mathrm{a}}_{u,d}(k) = \left(\frac{T_{\mathrm{s}} - \gamma(k)}{T_{\mathrm{s}}}\right)\cdot \sum_{i_m \in I_{u,d}} m^{\mathrm{l}}_{i_m,u,d}(k - \tau(k)) + \left(\frac{\gamma(k)}{T_{\mathrm{s}}}\right)\cdot \sum_{i_m \in I_{u,d}} m^{\mathrm{l}}_{i_m,u,d}(k - \tau(k) - 1),$$

where

$$\tau(k) = \mathrm{floor}\left\{\frac{\big(C_{u,d} - q_{u,d}(k)\big)\cdot l_{\mathrm{veh}}}{N^{\mathrm{lane}}_{u,d}\cdot v^{\mathrm{free}}_{u,d}\cdot T_{\mathrm{s}}}\right\},$$

$$\gamma(k) = \mathrm{rem}\left\{\frac{\big(C_{u,d} - q_{u,d}(k)\big)\cdot l_{\mathrm{veh}}}{N^{\mathrm{lane}}_{u,d}\cdot v^{\mathrm{free}}_{u,d}\cdot T_{\mathrm{s}}}\right\},$$

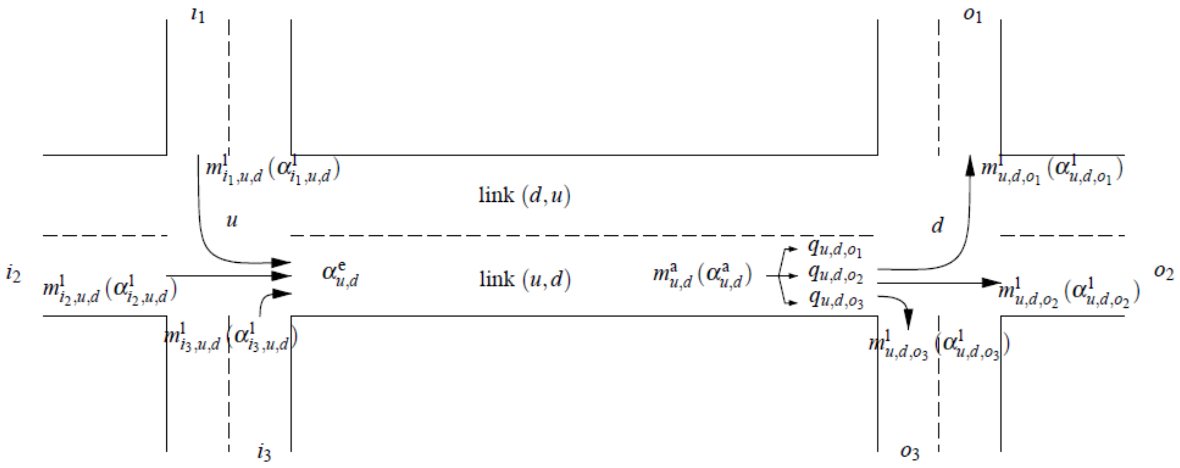


Fig. 1. A link connecting two traffic-signal-controlled intersections

with floor(x) referring to the largest integer smaller than or equal to x, and rem(x) is the remainder. The fraction of the arriving traffic in link (u,d) turning to $o_m \in O_{u,d}$ is

$$m^{\mathrm{a}}_{u,d,o_m}(k) = \beta_{u,d,o_m}(k) \cdot m^{\mathrm{a}}_{u,d}(k) \quad .$$

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

$$q_{u,d,o_m}(k+1) = q_{u,d,o_m}(k) + m^{\mathrm{a}}_{u,d,o_m}(k) - m^{\mathrm{l}}_{u,d,o_m}(k)$$

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

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

The new available storage stage depends on the number of cars that enter and leave the link in the period $[kT_{\mathrm{s}}, (k+1)T_{\mathrm{s}})$:

$$S_{u,d}(k+1) = S_{u,d}(k) - \sum_{i_m \in I_{u,d}} m^{\mathrm{l}}_{i_m,u,d}(k) + \sum_{o_m \in O_{u,d}} m^{\mathrm{l}}_{u,d,o_m}(k) \; .$$

2.2 *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. 2 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

$$n_{u,d}(k_d+1) = n_{u,d}(k_d) + \left(\alpha^{\mathrm{e}}_{u,d}(k_d) - \alpha^{\mathrm{l}}_{u,d}(k_d)\right) \cdot c_d \quad . \qquad (1)$$

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

$$\alpha^{\mathrm{l}}_{u,d}(k_d) = \sum_{o_m \in O_{u,d}} \alpha^{\mathrm{l}}_{u,d,o_m}(k_d), \quad o_m \in O_{u,d} \quad . \qquad (2)$$

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:

$$\begin{aligned}\alpha^{\mathrm{l}}_{u,d,o_m}(k_d) = \min\Big(&\beta_{u,d,o_m}(k_d) \cdot \mu_{u,d} \cdot g_{u,d,o_m}(k_d)/c_d, \\ &q_{u,d,o_m}(k_d)/c_d + \alpha^{\mathrm{a}}_{u,d,o_m}(k_d), \\ &\beta_{u,d,o_m}(k_d)\,(C_{o_m} - n_{o_m}(k_d))/c_d\Big) \quad .\end{aligned} \qquad (3)$$

The number of vehicles waiting in the queue turning to link o_m is updated as

$$\begin{aligned}q_{u,d,o_m}(k_d+1) = {} & q_{u,d,o_m}(k_d) + \\ & \left(\alpha^{\mathrm{a}}_{u,d,o_m}(k_d) - \alpha^{\mathrm{l}}_{u,d,o_m}(k_d)\right) \cdot c_d \quad .\end{aligned} \qquad (4)$$

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

$$q_{u,d}(k_d) = \sum_{o_m \in O_{u,d}} q_{u,d,o_m}(k_d) \quad . \qquad (5)$$

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

$$\begin{aligned}\alpha^{\mathrm{a}}_{u,d}(k_d) = {} & (1-\gamma(k_d)) \cdot \alpha^{\mathrm{e}}_{u,d}(k_d - \tau(k_d)) + \\ & \gamma(k_d) \cdot \alpha^{\mathrm{e}}_{u,d}(k_d - \tau(k_d) - 1),\end{aligned} \qquad (6)$$

$$\tau(k_d) = \mathrm{floor}\left\{\frac{(C_{u,d} - q_{u,d}(k_d)) \cdot l_{\mathrm{veh}}}{N^{\mathrm{lane}}_{u,d} \cdot v^{\mathrm{free}}_{u,d} \cdot c_d}\right\},$$

$$\gamma(k_d) = \mathrm{rem}\left\{\frac{(C_{u,d} - q_{u,d}(k_d)) \cdot l_{\mathrm{veh}}}{N^{\mathrm{lane}}_{u,d} \cdot v^{\mathrm{free}}_{u,d} \cdot c_d}\right\}. \qquad (7)$$

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:

$$\alpha^{\mathrm{a}}_{u,d,o_m}(k_d) = \beta_{u,d,o_m}(k_d) \cdot \alpha^{\mathrm{a}}_{u,d}(k_d). \qquad (8)$$

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

$$\alpha^{\mathrm{e}}_{u,d}(k_d) = \sum_{i_m \in I_{u,d}} \alpha^{\mathrm{l}}_{i_m,u,d}(k_d). \qquad (9)$$

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

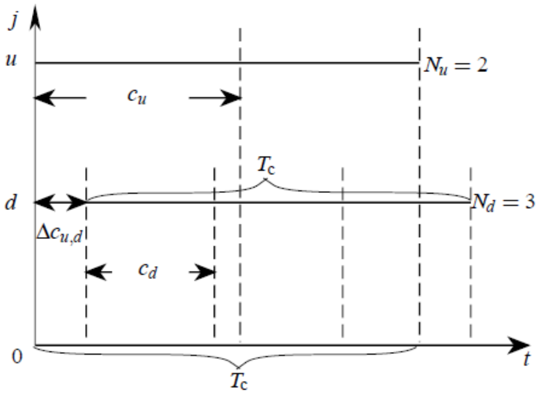


Fig. 2. Relationship between cycle times and control time interval

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.

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

$$T_c = N_j \cdot c_j, \quad \text{for } j \in J \tag{10}$$

with N_j an integer.

So T_c is the least common multiple of all the intersection cycle times in the traffic network. As Fig. 2 shows, we have

$$T_c = N_u \cdot c_u = N_d \cdot c_d. \tag{11}$$

For a given k_c the simulation time step counters for both intersections can range as follows:

$$\begin{aligned} k_u &= N_u \cdot k_c + p_u, \quad & p_u &= 0, 1, \ldots, N_u - 1 \\ k_d &= N_d \cdot k_c + p_d, \quad & p_d &= 0, 1, \ldots, N_d - 1. \end{aligned} \tag{12}$$

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

$$\alpha^1_{i_m,u,d}(t) = \alpha^1_{i_m,u,d}(k_u), \quad k_u \cdot c_u \le t < (k_u + 1) \cdot c_u, \tag{13}$$

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. 3 shows:

$$\alpha^1_{i_m,u,d}(k_d) = \frac{\displaystyle\int_{k_d \cdot c_d + \Delta c_{u,d}}^{(k_d+1)\cdot c_d + \Delta c_{u,d}} \alpha^1_{i_m,u,d}(t)}{c_d}\, \mathrm{d}t \ . \tag{14}$$

3. SIMULATION EXPERIMENTS

In centralized MPC, a fast running traffic network model is needed to satisfy the on-line optimization requirements. So, simulations are designed and carried out to verify whether the new simplified model (S model) can save time compared with the more detailed model (BLX model) while retaining a sufficiently high level of accuracy. The two models are compared for different network input flow rates, different prediction horizons, and different traffic network scales. During the experiment, the simulation time interval of the BLX model is set to 1 s, while the simulation time intervals of the S model are cycle times

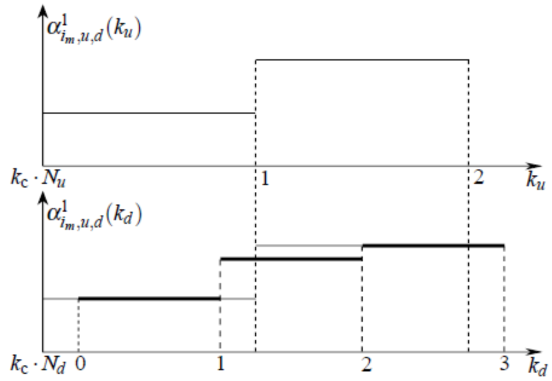


Fig. 3. Illustration for synchronizing flow rates

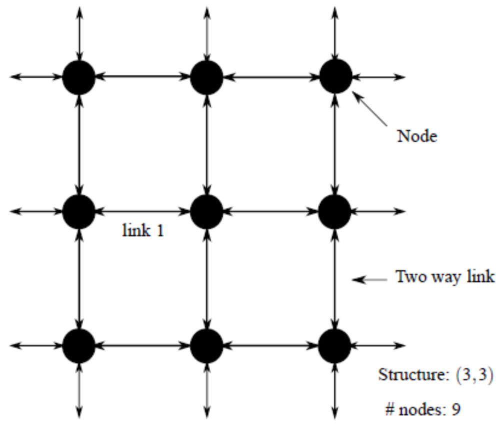


Fig. 4. The layout of a urban traffic network

which are 120 s, the same for all intersections in the network. The prediction horizons and traffic network scales are listed in Table 1.

Table 1. Traffic network characteristics and prediction horizon for each of the 5 simulation cases

Case number		1	2	3	4	5
Network	Structure	(1,2)	(3,3)	(8,8)	(13,13)	(18,18)
	# nodes	2	9	64	169	324
N_p		5	10	20	30	40

Each network considered is a grid-like network, where the "Structure" of the network is expressed as the number of nodes in each row and each column, and "# nodes" indicates the number of nodes. For example, Fig. 4 shows the layout of a $(3,3)$ network containing 9 nodes. "N_p" is the number of the control time intervals the model will run (i.e., simulation or prediction horizon expressed in steps of length T_c).

When using network 3, and $N_p = 10$, the computing times of the two models under different network input flow rates are shown in Fig. 5. The figure shows that the computing times are almost independent of the network input flow rates for both models. This means the traffic scenarios almost do not have any influence on the running time. Moreover, we can see from the figure that the S model required a much shorter computation time, around 0.5 s, while the BLX model took about 7 s, which is 14 times longer.

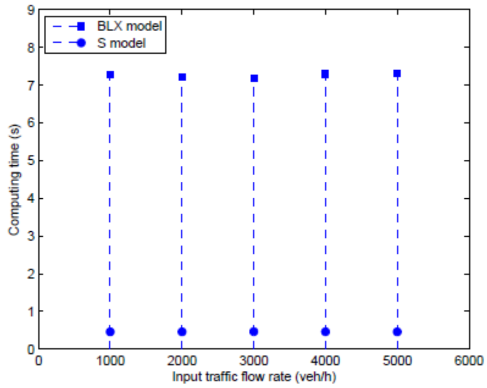


Fig. 5. The computing time consumed for different input flow rates of the traffic network

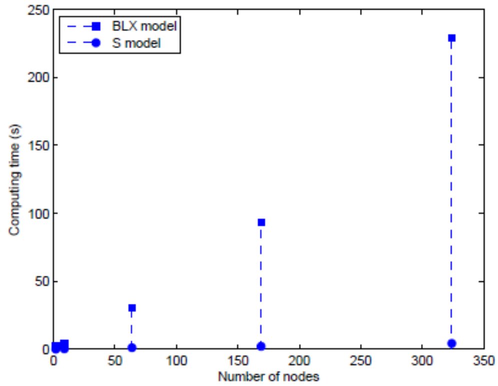


Fig. 7. The computing time consumed for different traffic network scales

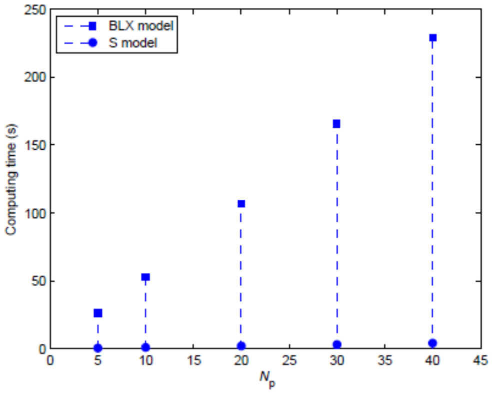


Fig. 6. The computing time consumed for different prediction horizons

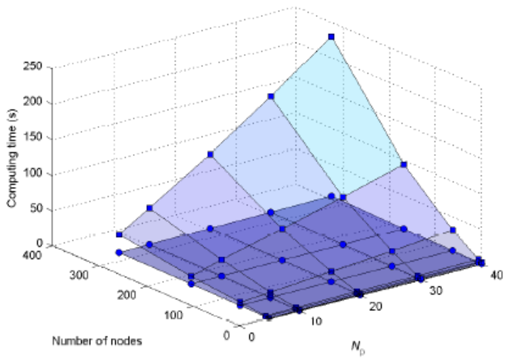


Fig. 8. The computing time consumed for both different prediction horizons and different traffic network scales

In each step of MPC for traffic control, a numerical optimization problem needs to be solved to obtain the optimal input value for the next step (using, e.g., a multi-start Sequence Quadratic Programming (SQP) algorithm). During the optimization, the model may need run hundreds to thousands of times. Therefore, by decreasing the computing time of the model, the on-line optimization time in MPC can be dramatically reduced.

Fig. 6 shows the changing of the running time with N_p, when the traffic network is set to network 5. Fig. 7 shows the changing of the running time with network scale, when $N_p = 40$. From the two figures, we can see that the longer the model is predicting, the larger scale the network is set to, the more time that the S model will save. The same conclusions can also be drawn from Fig. 8.

The S model is much faster than the BLX model, especially for longer prediction horizons N_p and larger network scales, but this extra speed is obtained by ignoring some details when modeling. Therefore, we need to verify whether the S model can still satisfy the requirements of control. The number of leaving vehicles can reflect the control effect of traffic lights on urban traffic, and Total Time Spent (TTS) is usually used as the control performance. If the S model shows behavior that is

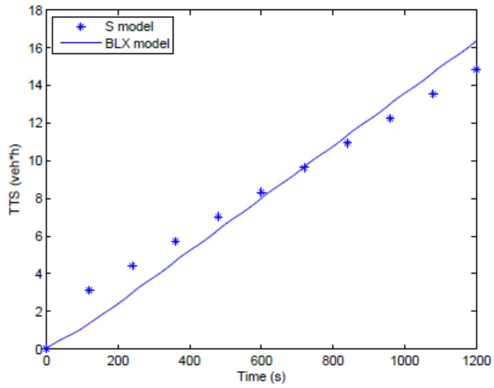


Fig. 9. The TTS for two models

similar to that of the BLX model for these two indexes, then it can be used as urban traffic control model guaranteeing similar control effects but with less control efforts. Fig. 9 and 10 are drawn for link 1 of network 2 (see Fig. 4), and $N_p = 10$. The figures show that the simplified model is accurate enough as a control model for urban traffic network.

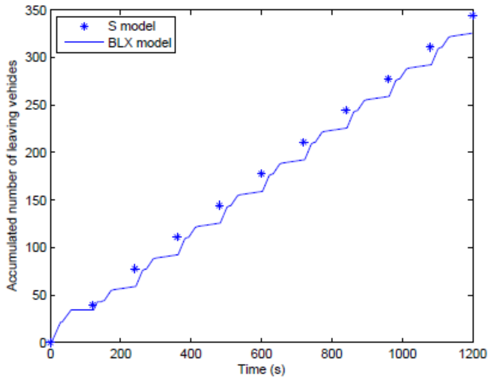


Fig. 10. The accumulated number of leaving vehicles for two models

4. CONCLUSIONS

A simplified macroscopic model has been established for controlling urban traffic network using model predictive control (MPC). This model takes the cycle times of the intersections as simulation time steps, where every intersection can have a different simulation time step. A control time interval, which is the least common multiply of all the cycle times, is defined to guarantee the communication and synchronization in the urban traffic network. The simplified model also describes how to ensure communication and synchronization between intersections with different simulation time steps.

The simplified model can take all typical traffic scenarios (saturated, unsaturated, and over-saturated traffic) into consideration, and is more flexible by having different cycle times. Moreover, it significantly reduces the computing time, which make it possible to be used for controlling larger urban traffic network.

However, the increasing of computing speed is obtained by enlarging the simulation time interval, which makes it lose some details and sacrifice some accuracy at the same time. But simulation results show that it guarantees enough accuracy to be used as the control model for urban traffic network.

Further research will focus on developing MPC algorithm to control urban traffic network based on this model, as well as an extensive assessment and comparison of the simplified model with a wide range of other traffic models for various network layouts and traffic demands when used for MPC-based traffic control.

ACKNOWLEDGEMENTS

This research is supported by a Chinese Scholarship Council (CSC) grant, the National Science Foundation of China (Grant No. 60674041), the Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20070248004), the European COST Action TU0702, the BSIK projects "Transition to Sustainable Mobility (TRANSUMO)" and "Next Generation Infrastructures (NGI)', the Delft Research Center Next Generation Infrastructures, and the Transport Research Centre Delft.

REFERENCES

Barisone, A., Giglio, D., Minciardi, R., and Poggi, R. (2002). A macroscopic traffic model for real-time optimization of signalized urban areas. In *Proc. of the 41st IEEE Conference on Decision and Control*, 900–903. Las Vegas, USA.

Diakaki, C., Papageorgiou, M., and Aboudolas, K. (2002). A multivariable regulator approach to traffic-responsive network-wide signal control. *Control Engineering Practice*, 10(2), 183–195.

Dotoli, M., Fanti, M.P., and Meloni, C. (2006). A signal timing plan formulation for urban traffic control. *Control Engineering Practice*, 14(11), 1297–1311.

Gazis, D.C. and Potts, R.B. (1963). The oversaturated intersection. In *Proc. of the 2nd International Symposium on Traffic Theory*.

Hegyi, A. (2004). *Model Predictive Control for Integrating Traffic Control Measures*. Ph.D. thesis, Delft University of Technology.

Kachroo, P. and Özbay, K. (1999). *Feedback Control Theory for Dynamic Traffic Assignment*. Advances in Industrial Control. Springer.

Kashani, H. and Saridis, G. (1983). Intelligent control for urban traffic systems. *Automatica*, 19(2), 191–197.

Lin, S. and Xi, Y. (2008). An efficient model for urban traffic network control. In *Proc. of the 17th World Congress the International Federation of Automatic Control*, 14066–14071. Seoul, Korea.

Lin, S., Xi, Y., and Yang, Y. (2008). Short-term traffic flow forecasting using macroscopic urban traffic network model. In *Proc. of the 11th International IEEE Conference on Intelligent Transportation Systems*, 134–138. Beijing, China.

Papageorgiou, M. (1983). *Applications of Automatic Control Concepts to Traffic Flow Modeling and Control*. Lecture Notes in Control and Information Sciences. Springer Verlag, Berlin, Germany.

van den Berg, M., De Schutter, B., Hegyi, A., and Hellendoorn, J. (2004). Model predictive control for mixed urban and freeway networks. In *Proc. of the 83rd Annual Meeting of the Transportation Research Board*, 19. Washington, D.C.

van den Berg, M., Hegyi, A., De Schutter, B., and Hellendoorn, J. (2003). A macroscopic traffic flow model for integrated control of freeway and urban traffic networks. In *Proc. of the 42nd IEEE Conference on Decision and Control*. Maui, Hawaii USA.

Appendix B

S-Model Manual

The S-model is an executable file in the format:

```
mexLM.mexw32
```

and can be executed within Matlab by calling:

```
mexLM(input arguments)
```

The number of input arguments depends on what the S-model should do (initialize network, calculate traffic, update network status, delete network). It can be derived from the file mexLM.cpp what will happen with which input arguments. All the modes used in the Matlab code are always explained in the comments. Also, information on the format of the input arguments can be found in both Matlab commentary and the original C++ code.

To get the S-Model running, the following software should be installed:

- Qt3 libraries for Windows
- Microsoft Visual Studio 2005 and Service Pack 1
- Matlab 2008b

The S-model is written in C++ and can be compiled in both Matlab and Visual Studio. This manual will explain the procedure for Visual Studio step by step. It is assumed the operation system is Windows XP, for other versions of Windows no guarantees about the correct working of the software can be given. Usually a compiled and working mexLM.mexw32 file will work on other computers (given the three software packages are correctly installed), but it is not uncommon that it will have to be recompiled to get it working.

Whenever in this manual is referred to a path with dots, for example:

c:\...\qt3\bin

Please fill in the relevant directory, as this can be different for each computer.

Microsoft Visual Studio 2005

The first step is installing Visual Studio 2005. It has not been tested of newer versions of Visual Studio will perform correctly. Make sure the

C++ module is installed. Next, install service pack 1 (SP1), which is a rather large file (431.7Mb). It can be downloaded from:

http://www.microsoft.com/downloads/details.aspx?FamilyId=BB4A75AB-E2D4-4C96-B39D-37BAF6B5B1DC&displaylang=en

Also, download and install the latest available security patches from the same website.

Qt3 libraries for Windows

These libraries are not officially available for Windows (only Linux and Mac), but have been made available for Windows by the qtwin-project. They are open source and can be retrieved from the dvd enclosed by this report or downloaded from the following source:

http://sourceforge.net/projects/qtwin/files/Unofficial%20Qtwin/

The files should be just unzipped or copied, nothing needs to be installed (yet). It is absolutely vital that you install this version (3.3.x), **both newer and older versions of Qt libraries will be incompatible**! The next steps are taken from the following source:

http://qtwin.sourceforge.net/qt3-win32/compile-msvc.php

Then, the Qt3-libraries need to be installed in Visual Studio. To do so, open the Command Prompt for Visual Studio and execute the following commands:

```
c:\...\Microsoft Visual Studio\VC\bin\VCVARS32.BAT
set QTDIR=< qt3 source root >
set PATH=%QTDIR%\bin;%PATH%
set QMAKESPEC=win32-msvc
c:\...\qt3\configure-msvc-2005.bat -fast
```

The Command Prompt can be closed. Then in Windows, go to Control Panel → System → Advanced → Environment Variables → Path
In "Path", add the following line:

```
c:\...\qt3\bin;
```

Matlab 2008b

Install Matlab 2008b. Other versions could work as well, this has not been tested and can therefore not be guaranteed. After installation, the first step is to set the MEX-function to the Visual C++-compiler. To do so enter the following commands in Matlab:

```
mex –setup
```

Choose the relevant compiler by its number, confirm with Y(es)

To check if the C++-compiler works correctly, try to compile the file mexcpp.cpp. To do so, set the active folder to:

```
C:\...\MATLAB\R2008b\extern\examples\mex
```

and type:

```
mex mexcpp.cpp
```

Then you can execute the function by:

```
mexcpp(3,4)
```

or any choice of two numbers to verify if it works correctly. If so, continue to the next step.

Microsoft Visual Studio 2005

Open Visual Studio and open the S-model solution: mexLM.sln
Then, the Qt3 and Matlab libraries need to be made explicit to Visual Studio. Go to:
Tools → Options → Projects and Solutions → VC++ directories
For "include files", add:

```
C:\...\MATLAB\R2008b\extern\include
C:\...\qt3\include
```

For ""library files", add:

```
C:\...\MATLAB\R2008b\extern\lib and  C:\...\qt3\lib
```

For "source files", add:

```
C:\...\MATLAB\R2008b\extern\src and C:\...\qt3\src
```

In the Solution Explorer (the left part in Visual Studio), add the following file to "Resource files" (by drag-and-drop):

```
C:\Program
Files\MATLAB\R2008b\extern\include\mexversion.rc
```

Then, right-click on MexLM → Properties → C++ → General
In the field "Additional include directories", make sure the path to the Matlab directory is correct.

MexLM → Properties → Linker → General
In the field "Additional Library directories", make sure the path to the Matlab directory is correct.

MexLM → Properties → Linker → Input

For "Additional dependencies", check both the Matlab and Qt paths and correct them if necessary.

In the solution explorer, open the file "stdafx.h" and correct the following lines to their respective paths:

```
#include </../qt3/include/qptrlist.h>
#include </../qt3/include/qglobal.h>
#include </../qt3/include/qptrcollection.h>
```

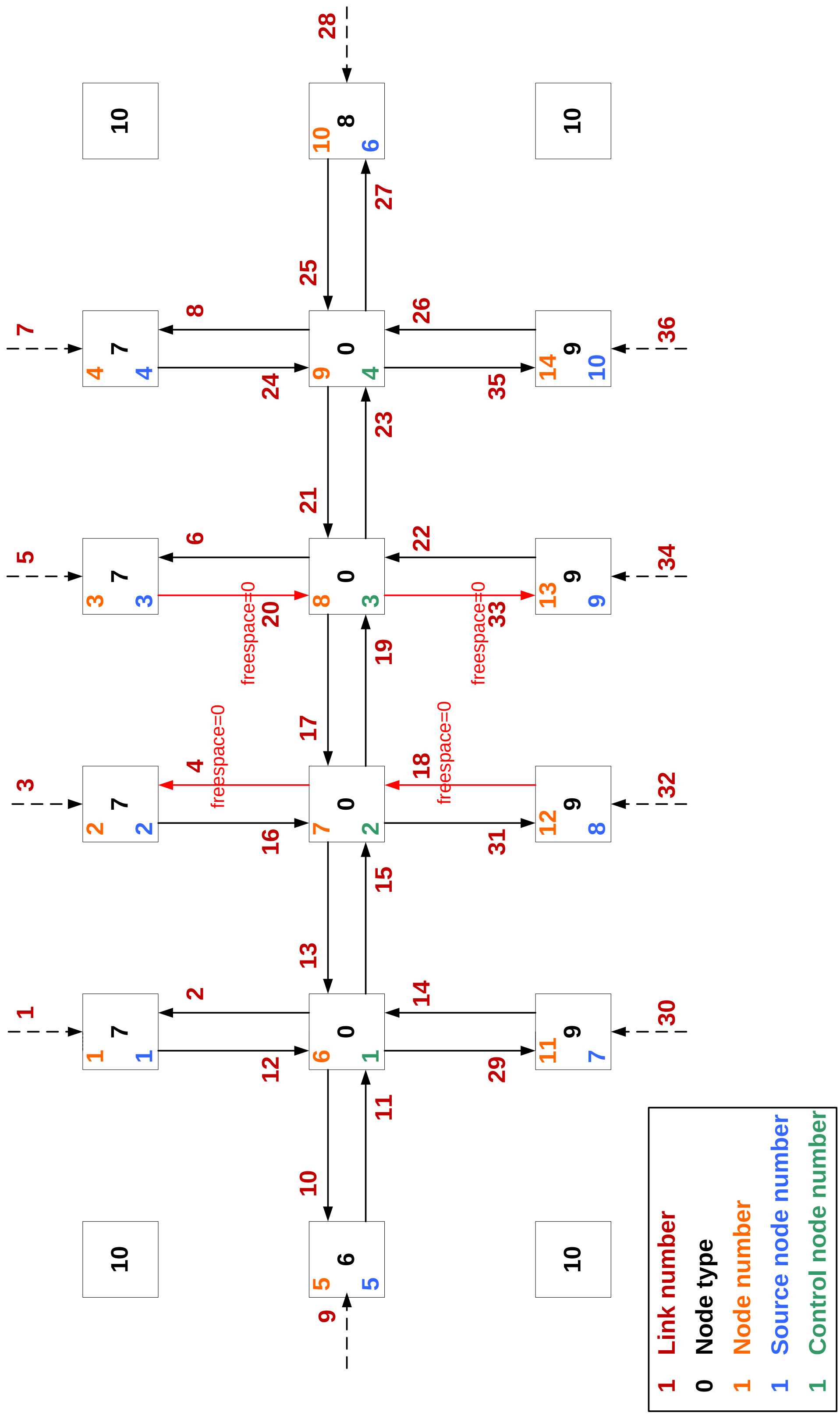
In mexLM.cpp, correct the following line if necessary:

```
#include "C:\...\MATLAB\R2008b\extern\include\mex.h"
```

Now, the S-model is ready to be compiled. Therefore, right-click "Build" or "Rebuild" on the solution. The following document can be used for general help on compiling Matlab mex-functions in Visual Studio:
http://www.saliencytoolbox.net/mexw64/CompileWin64.pdf

Appendix C

Node and link numbering in S-model



Appendix D

Number of runs, sample data

For ten runs, the travel time is measured. The travel time is in seconds.

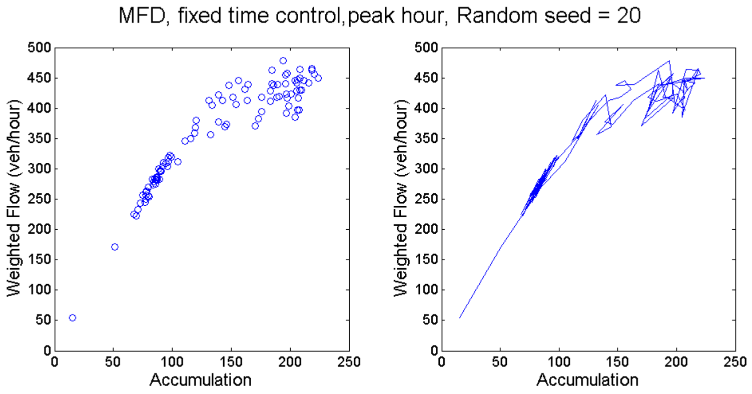
Time/Run	1	2	3	4	5	6	7	8	9	10
90	0	0	81	0	0	0	0	0	0	0
180	261	475	257	434	413	413	77	133	95	406
270	242	492	305	418	184	480	585	209	278	261
360	294	418	722	223	258	473	437	536	386	405
450	440	356	133	357	432	409	295	159	181	999
540	174	682	762	259	547	568	271	356	414	337
630	516	325	500	268	194	253	581	225	459	487
720	48	258	0	592	265	489	647	304	337	198
810	155	182	547	155	222	416	254	297	183	0
900	228	232	575	689	170	190	708	255	469	690
Total	2358	3419	3882	3393	2684	3691	3855	2474	2802	3784

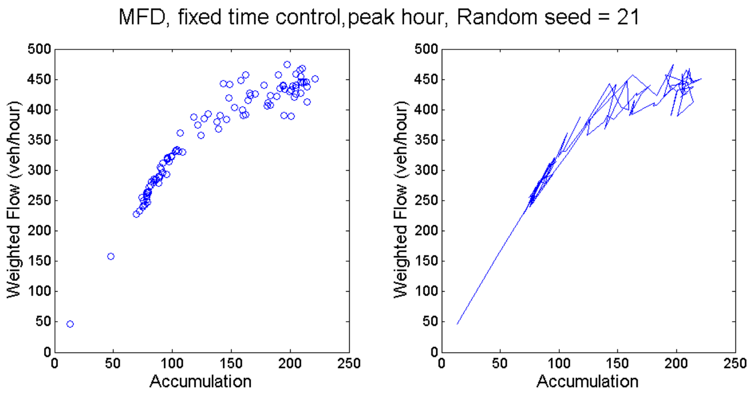
	95%	90%
Mean (of the samples)	3.234	3.234
St. Dev (of the samples)	597	597
Acceptable St dev	741	494
T value	1,812	1,372
N	5,0	5,0

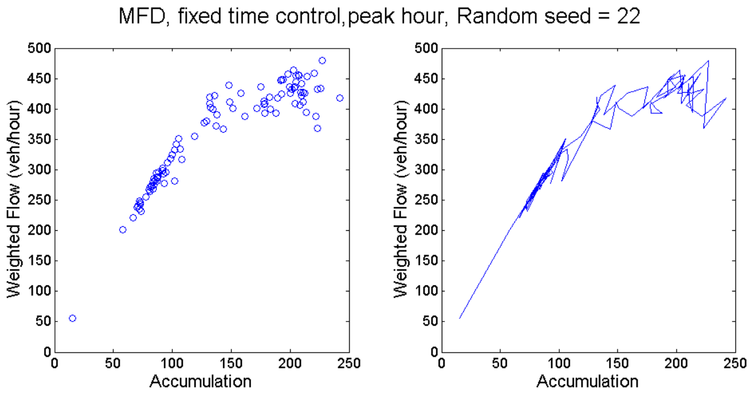
Appendix E

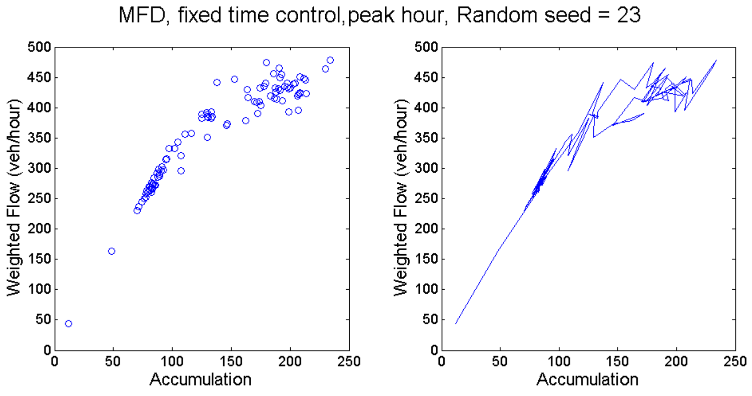
MFD's Peak hour

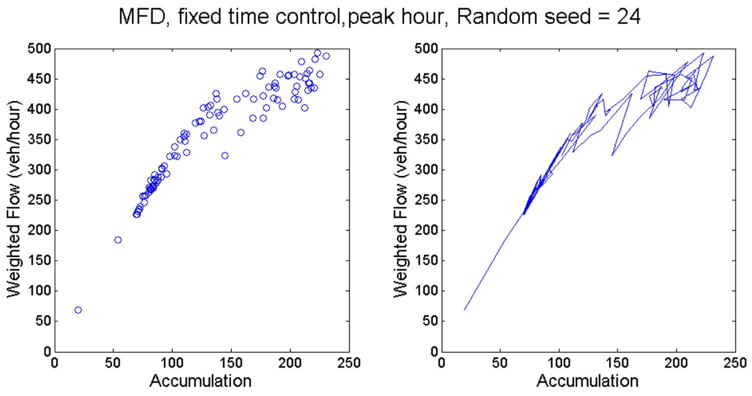
Fixed time control



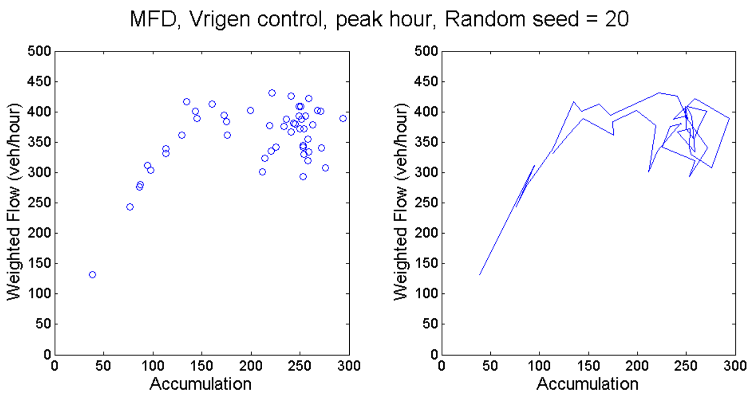


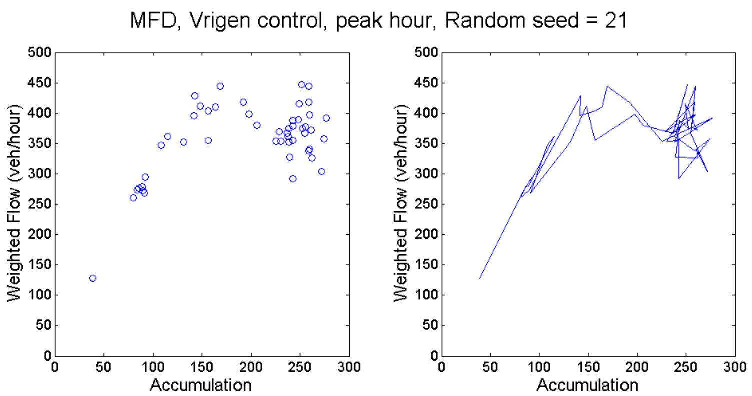


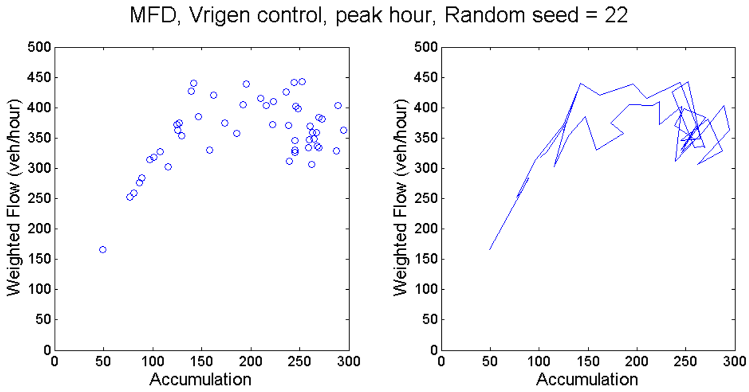


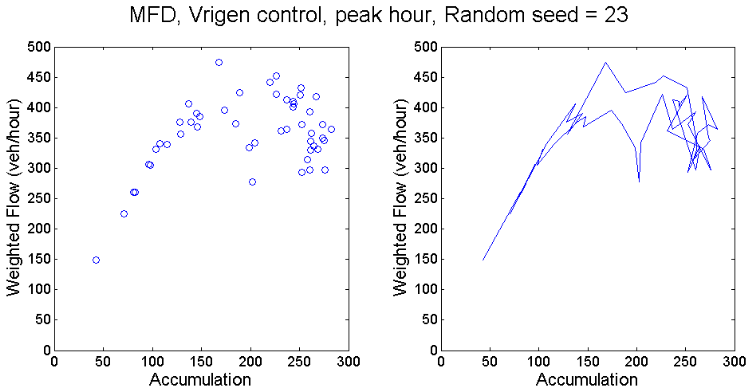


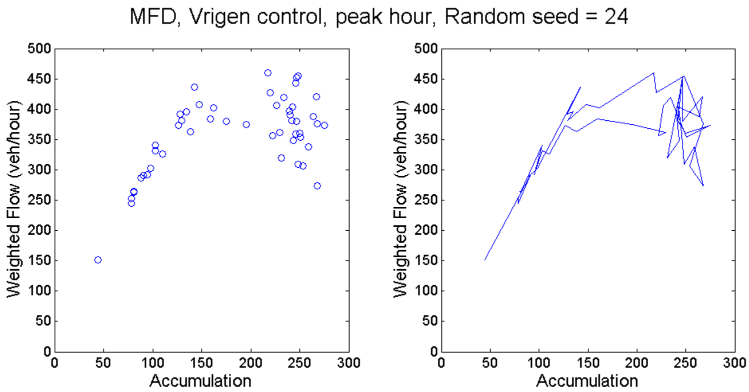
Vrigen control






MFD, Vrigen control, peak hour, Random seed = 22
Weighted Flow (veh/hour)
Accumulation
Weighted Flow (veh/hour)
Accumulation


MFD, Vrigen control, peak hour, Random seed = 23
Weighted Flow (veh/hour)
Accumulation
Weighted Flow (veh/hour)
Accumulation


MFD, Vrigen control, peak hour, Random seed = 24
Weighted Flow (veh/hour)
Accumulation
Weighted Flow (veh/hour)
Accumulation