CHALMERS

Chalmers University of Technology
Department of Signals and Systems

Bachelor Thesis SSYX02-16-22
January 5, 2017

Control Oriented Design of Variable Message Signs

SSYX02–16–22

Authors:
Lowisa Hanning
lowisa@student.chalmers.se
Gabriella Lygnestrand
gablyg@student.chalmers.se
Parasto Mohammadi
parmoh@student.chalmers.se
Per Ohlsson
peroh@student.chalmers.se
Vanessa Olsson
vanols@student.chalmers.se

Examiner:
Balázs Kulcsár
Supervisor:
Azita Dabiri

Control Oriented Design of Variable Message Signs
© Lowisa Hanning, Gabriella Lygnestrand, Parasto Mohammadi, Per Ohlsson, Vanessa Olsson, January 5, 2017.

Supervisor: Azita Dabiri, Department of Signals and Systems
Examiner: Balázs Kulcsár, Department of Signals and Systems

Bachelor Thesis 2016
Department of Signals and Systems
Chalmers University of Technology
SE-412 96 Gothenburg

Gothenburg, Sweden January 5, 2017

SAMMANDRAG

Variabla meddelande skyltar, tillsammans med variabla hastighetsgränser, används idag i stor utsräckning i Europa men i relativt låg utsträckning i Sverige. Syftet med dessa skyltar är bland annat att skapa ett jämnare trafikflöde och undvika trafikstockningar på motorvägar. I många delar av Sverige är regulatordesignen för de variabla hastighetsgränserna utdaterad och i stort behov av en förbättring. Detta kandidatarbete undersöker potentialen till en vidare utveckling och implemetering av en optimerad algoritm och regulatordesign för variabla hastighetsgränser med syfte att uppnå ett jämnare trafikflöde. Andra metoder för att förbättra trafikflödet, som till exempel förändringar i vägens infrastruktur eller rampmätning, har inte undersökts. För att ta reda på vilka förbättringsmöjligheter som kan utföras implementerades och analyserades den makroskopiska modellen METANET i MATLAB. Modellen användes sedan tillsammans med en mikroskopisk modell som skapades i trafiksimuleringsprogrammet VISSIM. Den makroskopiska och den mikroskopiska modellen kombinerades för att kunna analysera trafiken både som ett flöde och som ett system av individulla fordon. Modellerna används för att kunna skapa en skräddarsydd regulatordesign som ändrar hastighetsgränserna med avseende att optimera trafikflödet. Regulatorn applicerades och testades på den simulerade vägen både i den mikroskopiska modellen och den makroskopiska modellen med olika scenarion och indata, som till exempel fordonsinput under olika tidsintervall. Tre olika scenarion valdes ut för att testa om implementering av variabla hastigetsgränser gav önskad effekt. Resultatet blev ett förbättrat trafikflöde när variabla hasighetsgränser var implementerade, jämfört med när de inte var det.

ABSTRACT

Variable message signs and variable speed limits are today frequently used in many parts of Europe but is not as well explored in Sweden. The purpose of these signs is e.g. to create a more homogeneous traffic flow and prevent congestion on freeways. In many places of Sweden the control algorithm is outdated and in need of a revision. In this research the potential of further development and implementation of an optimized control algorithm and controller design for variable speed limits is investigated. Other methods to improve the traffic flow, for example ramp metering or changing the infrastructure of the road, has not been examined. To research the potential and develop a controller design of variable message signs, with variable speed limits, firstly the microscopic model called METANET was implemented and analyzed in MATLAB. This model was then used with a microscopic model created in the software VISSIM. The macroscopic and the microscopic models were combined to be able to analyze the traffic both as a flow and as a system of individually simulated vehicles. The two models were used to create a custom-made control design which changes the speed limits with the aim at optimizing the traffic flow. The controller were applied and tested on the simulated roads, both in the microscopic model and in the macroscopic model with different scenarios and input data, regarding e.g. vehicle inputs during different time intervals. Three different scenarios was chosen to test the effect given when variable speed limits was applied, compared to the cases without variable speed limits. The result was that the case when variable speed limits was applied improved the traffic flow for all the tested scenarios.

ACKNOWLEDGEMENTS

The work and process reported in this thesis has been supervised by Azita Dabiri and examined by Balázs Kulcsár from the department of Signals and Systems at Chalmers University of Technology in Gothenburg. Both have been extraordinary supervisors and have given us a lot of support, ideas and good critique in the thesis work. It has been a great collaboration and we hope that we will get the opportunity to work with them again and expand our relation in the future. We would also want to send a big thank you to Balázs Varga from the Corvinus University of Budapest for helping us in answering our questions, solving problems and understanding the VISSIM software and COM interface. We would also like to thank the Swedish Transport Administration for letting us execute a study visit and get the chance to see how they use VSL and VMS today. From the Swedish Transport Administration we would like to thank Pernilla Fransson and Mats Olsson for meeting us and showing us around the Swedish Transport Administration office and answering our general questions. For answering our more technical questions about the use in Gothenburg today we would like to thank Emil Muda.

Contents

1 Introduction **1**
- 1.1 Purpose 2
- 1.2 Scope 2
- 1.3 Method 2
  - 1.3.1 Literature study 2
  - 1.3.2 Modeling in MATLAB 2
  - 1.3.3 Orientation in simulation software and implementing COM interface 3
  - 1.3.4 Controller design 3
  - 1.3.5 Study visit/case study 3
  - 1.3.6 Evaluation 3
- 1.4 Outline of the report 3

2 Traffic flow models **4**
- 2.1 METANET 4
  - 2.1.1 Link equations 5
  - 2.1.2 The fundamental diagram 6
  - 2.1.3 On-ramp equations 6
  - 2.1.4 Boundary conditions 7
- 2.2 Cremer's models 8
- 2.3 Modeling congestion 8

3 Traffic control measures **13**
- 3.1 Variable message signs 13
- 3.2 Variable speed limits 13
- 3.3 Combining VMS and VSL 14
- 3.4 Other control measures 14
- 3.5 Coordination and design of VMS controller 15
  - 3.5.1 Model Predictive Control 15
- 3.6 Traffic control in Gothenburg 15
- 3.7 Use of VMS and VSL in Europe today 18

4 Traffic simulation in VISSIM **19**
- 4.1 Network objects in VISSIM 19
- 4.2 Simulation 20
- 4.3 Evaluate simulation data 21
- 4.4 VISSIM COM Interface 21

5 Model calibration **23**
- 5.1 Parameter fitting 23
- 5.2 Minimization method fmincon 24

6 Controller design **26**
- 6.1 Linearization of METANET 26
- 6.2 Controller Implementation 27
- 6.3 Filtering 27
- 6.4 Controller in METANET 27
- 6.5 Controller in COM interface 28

7 Implementation and results **29**
- 7.1 Mölndal E6/E20 29
  - 7.1.1 Collection of real time data 30
- 7.2 METANET - boundaries from VISSIM 31
  - 7.2.1 Calibration 32
  - 7.2.2 Validation 33

7.2.3 Scenario 1 35
7.3 METANET - selected boundary conditions 37
7.3.1 Scenario 2 37
7.3.2 Scenario 3 40
7.4 COM interface - VISSIM 42
7.4.1 Scenario 4 42
7.4.2 Scenario 5 49

8 Discussion 54

9 Conclusion 56

References 56

Appendices 58

A METANET 58

B Load excel 61

C Minimization 62

D Objective function 63

E Linearization 65

F Steady state 67

G COM interface 68

LIST OF ACRONYMS/ABBREVIATIONS

Abbreviation	Full name
DSL	Dynamic Speed Limits
ITS	Intelligent Transportation Systems
LASSY	Lane Signaling System
LQR	Linear-Quadratic Regulator
MCS	Motorway Control System
MPC	Model Predictive Control
NTS	National Traffic System
TCM	Traffic Control Measures
VISSIM	Verkher In Städten- SIMutaltionsmodell
VMS	Variable Message Signs
VSL	Variable Speed Limits

List of notations

Notation	Explanation
m	Subscript m denotes a motorway link
i	Subscript i denotes a segment
$\rho_{m,i}$	Traffic density [veh/km/lane]
$v_{m,i}$	Mean speed [km/h]
$q_{m,i}$	Traffic volume or outflow [veh/h]
$V[\rho]$	Speed-density relationship [km/h]
T	Time step [s]
k	Time instant t=kT
K	Time horizon
N_m	Number of segments
L_m	Length of a link
λ_m	Number of lanes
v_{fm}	Average free speed [km/h]
τ	Relaxation time constant
ν	Anticipation constant
κ	Speed anticipation parameter
α_m	A nonlinear relation between mean speed and density
δ	Model parameter used when on-ramps are implemented
$\rho_{cr,m}$	Critical density [veh/km/lane]
ρ_{max}	Maximum density for the link [veh/km/lane]
F_{obj}	Objective function
w_o	Queue length at origin o
d_o	Demand flow at o [veh/h]
q_o	Ramp outflow [veh/h]
r_o	Ramp metering rate
Q_o	Ramp flow capacity [veh/h]
ρ_{max}	Maximum density of a segment [veh/km/lane]
u	Control signal
l	Design parameter
m	Design parameter
N_{samp}	Number of data samples
I_{all}	Number of all links and segments
$\tilde{q}_{m,i}$	Measured flow [veh/h]
$\tilde{v}_{m,i}$	Measured average speed [km/h]
ξ	Weight constant
$\hat{q}_{m,i}$	Simulated flow [veh/h]
$\hat{v}_{m,i}$	Simulated average speed [km/h]
$\boldsymbol{A}$	State matrix
$\boldsymbol{B}$	Input matrix
$\boldsymbol{C}$	Output matrix
$\boldsymbol{D}$	Feedthrough matrix
$\boldsymbol{x}$	Input vector
$\boldsymbol{y}$	Output vector
k_{lqr}	The gain output from the controller
v_{stat}	Stationary speed used in the linerization
V_{limit}	The stationary speed limit, before VSL is implemented
$G(s)$	Transfer function used as a filter
x_f	Input vector used for the filter
a_f	State matrix used for the filter
b_f	Input matrix used for the filter
u_f	Output used for the filter
c_f	Output matrix used for the filter
d_f	Feedthrough matrix used for the filter
τ_f	Time constant used for the filter

List of Figures

1.1 Variable message signs and variable speed limits (Highways England, 2013) . . 1
1.2 Process of the research 2
2.1 Segment i of motorway link m, with $\lambda_m = 2$ 5
2.2 An example of a fundamental diagram 6
2.3 Traffic conditions for a segment where an on-ramp is connected depends on the characteristics for the ramp and the segment. The queue length depends on the amount of vehicles that wants to enter the ramp and the amount of cars that is entering the segment and leaving the ramp. 7
2.4 Congestion propagating forward, without VSL 9
2.5 Congestion propagating forward, with VSL model 1 9
2.6 Congestion propagating forward, with VSL model 2 10
2.7 Congestion propagating backwards, without VSL 10
2.8 Congestion propagating backwards, with VSL model 1 11
2.9 Congestion propagating backwards, with VSL model 2 11
3.1 Overhead gantry sign combining VMS and VSL (David Dixon, 2013) 14
3.2 Recommended VSL and mandatory VSL 16
3.3 TV-screen at the Swedish Transport Administration that shows a VSL sign in Gothenburg 16
3.4 Control room at the Swedish Transport Administration 17
3.5 A screenshot from Trafiken.nu 17
4.1 How to change network objects in VISSIM 20
4.2 3D simulation in PTV VISSIM software 21
4.3 Simulation results evaluated in an excel file. 21
5.1 The calibration procedure 23
5.2 Example of the parameter fitting, the calibration is done by reducing the difference between the model and the real data. 24
7.1.1 A map of the selected area and road Mölndal E6/E20 29
7.1.2 An overview of the modeled road in Mölndal 30
7.2.1 Fundametal diagram for simulation 1 32
7.2.2 Density in segment 2, simulation run 1 in VISSIM and best calibrated model in METANET 33
7.2.3 Speed in segment 2, simulation run 1 in VISSIM and best calibrated model in METANET 33
7.2.4 Density in segment 2, simulation run 2 in VISSIM and the validated model in METANET 34
7.2.5 Speed in segment 2, simulation run 2 in VISSIM and the validated model in METANET 34
7.2.6 The control variable, $V_{controlled}$ and $V_{uncontrolled}$ 35
7.2.7 Density in segment 2, when implementing VSL 36
7.2.8 3D density plot, simulation run 1, data from VISSIM 36
7.2.9 3D density plot, uncontrolled case, calibrated model. 37
7.2.10 3D density plot, controlled case, calibrated model. 37
7.3.1 Density for the uncontrolled simulation when the congestion propagates forward, from a 3D view 38
7.3.2 Density for the controlled simulation when the congestion propagates forward, from a 3D view 39
7.3.3 Density in segment number 3 for the congestion that propagates forward 39
7.3.4 Speed limit in segment number 3, congestion propagating forward 40
7.3.5 Density for the uncontrolled simulation when the congestion propagates backwards, from a 3D view 41
7.3.6 Density for the controlled simulation when the congestion propagates backwards, from a 3D view 41
7.4.1 3D density plot, scenario 4, uncontrolled. 42

7.4.2 2D view density plot, scenario 4, uncontrolled. 43
7.4.3 3D density plot, scenario 4, controlled. 44
7.4.4 2D view density plot, scenario 4, controlled. 44
7.4.5 Density plot, scenario 4, segment 5. 45
7.4.6 3D mean speed plot, scenario 4, uncontrolled. 46
7.4.7 2D view mean speed plot, scenario 4, uncontrolled. 46
7.4.8 3D mean speed plot, scenario 4, controlled. 47
7.4.9 2D view mean speed plot, scenario 4, controlled. 47
7.4.10 Change in speed limit over time, in segment 3-6 48
7.4.11 3D density plot, scenario 5, uncontrolled. 49
7.4.12 2D view density plot, scenario 5, uncontrolled. 49
7.4.13 3D density plot, scenario 5, controlled. 50
7.4.14 2D view density plot, scenario 5, controlled. 50
7.4.15 Density plot, scenario 5, segment 4. 51
7.4.16 3D mean speed plot, scenario 5, uncontrolled. 52
7.4.17 2D view mean speed plot, scenario 5, uncontrolled. 52
7.4.18 3D mean speed plot, scenario 5, controlled. 53
7.4.19 2D view mean speed plot, scenario 5, controlled. 53

1 Introduction

Traffic congestion is a frequent problem in the current traffic networks. In our time studded society, people are expecting to get from one place to another smoothly without delays or complications. The occurrence of traffic congestion is often obstructing that and is a problem to be addressed and solved. One of many consequences of traffic congestion, besides the indisputable time problem, is the elevated risk of accidents. The instances of stationary vehicles and precipitous braking entails to an enlarged risk for collisions. This heightened risk can be decreased by properly controlling the traffic flow and speed. Another effect of traffic congestion is idle driving, which has a negative impact on the environment. With sustainability being a central part of an ongoing global debate, coming up with intelligent transportation systems (ITS) is of great importance in the process of decreasing emissions. With the help of ITS, traffic congestion can be compensated without changing the infrastructure (Hegyi, 2004).

Traffic congestion occurs when the traffic demand is too high for the network. One cause can be that the interaction between vehicles slows down the speed and creates a bottleneck effect. Another possible cause of traffic congestion can be an accident occurring upstream in the link decreasing the accessibility. An important motorway control strategy is Variable Message Signs (VMS) and Variable Speed Limits (VSL), see figure 1.1. VSL is also called dynamic speed limits (DSL). These strategies aims at regulating traffic speed in the motorways based on the changes in traffic conditions e.g. traffic accidents or traffic jam, as well as change in the weather conditions (heavy snow, rain or fog). To decide if the signs should be switched on or off, the majority of VSL has an algorithm based on some threshold values e.g. traffic flow, occupancy, mean speed or a combination of these values (Nygårdhs and Helmers, 2007).

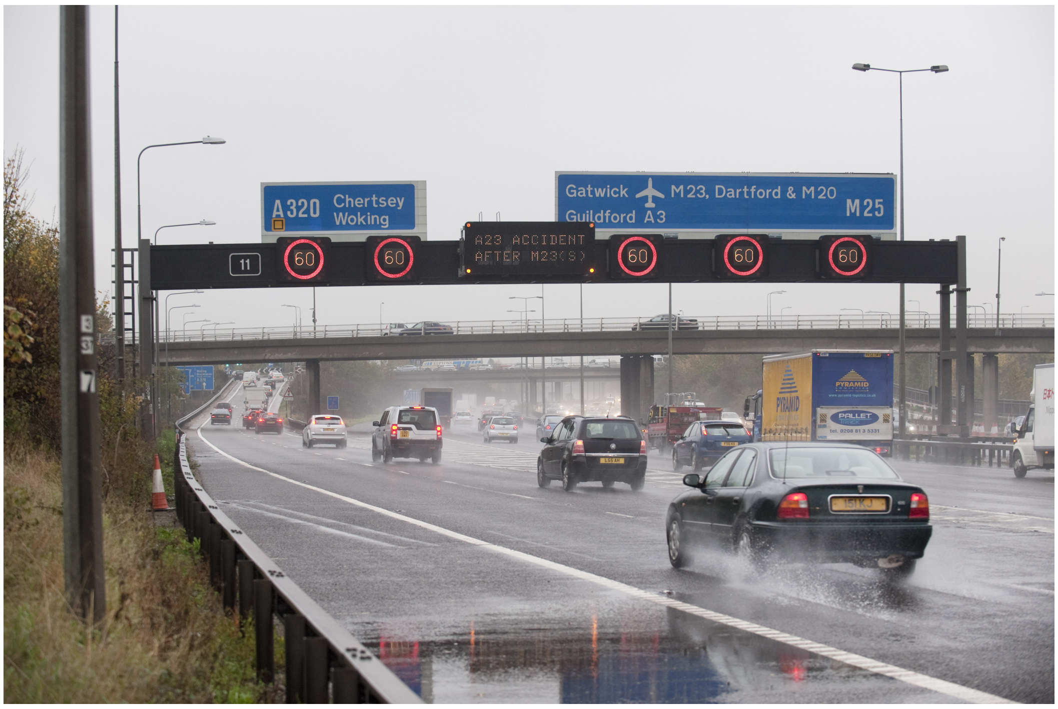


Figure 1.1: Variable message signs and variable speed limits (Highways England, 2013)

VMS and VSL is used in the traffic in European countries like Sweden, Netherlands, Finland and United Kingdom. However, each country has different data collecting devices and parameters used in their algorithms. Finland uses weather stations to gather data meanwhile in Netherlands loop detectors are being used. Sweden uses VSL signs on e.g. highway E4, but the regulation is only advisory (Abdel-Aty and Yu, 2014). Along the highway E4 in Sweden the recommended VSL signs can be found, wired on gantries that have microwave detectors to measure traffic volumes and speed. The speed threshold V_{low} and V_{high} are the main elements and is utilized to determine when the VSL system switch should be on or off, by comparing the observed and processed speeds to V_{low} and V_{high}. Some evaluation results implied that the VMS and VSL showed no significant impact on the traffic conditions on E4, which might be due to different reasons, e.g. the fact that the utilized VSL only is advisory (Nygårdhs and Helmers, 2007).

1.1 Purpose

Traffic congestion is a problem that causes loss of time in our daily life and has a negative impact on the environment. This report aims to describe the development of a control oriented design of VMS and VSL signs, by the use of a proper model or a more accurate control algorithm. The focus is on the highway E6/E20 in the Mölndal area, Gothenburg where 7 km of the highway is analyzed.

1.2 Scope

The project was time limited to a period of five months. The focus was on the traffic flow regarding the main freeways in Gothenburg and no other locations or road types will be taken into consideration. A further scope was done in the project where a segment of Gothenburg's traffic highway network was of focus. The study exclusively analyzed the design of VMS and VSL and does not regard any possible changes in the infrastructure and their effect on the traffic conditions.

1.3 Method

The report was divided into seven significant parts, see figure 1.2. All seven parts has played a major roll in the process. A literature study within the area of VMS and VSL was the first part that was performed. The second part was to to create a mathematical model in MATLAB. After that, an orientation in the simulation software PTV VISSIM was performed, to understand and learn how to manage the software. The fourth part of the process was to create a COM interface in VISSIM, to be able to implement speed limits in the simulation and to run VISSIM with MATLAB. The fifth part was to choose and design the controller and later implement the controller in the mathematical model and in the COM interface. A study visit to the Swedish Transport Administration in Gothenburg was also done, where a deeper knowledge for the traffic control strategies in Gothenburg was achieved. Information about the chosen road for the case study was also obtained. The last part consisted of evaluating and analyzing the results from different scenarios and the case study.

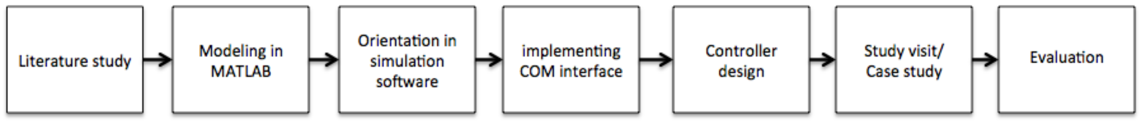


Figure 1.2: Process of the research

1.3.1 Literature study

Literature relevant for the topic have been reviewed and analyzed. A collocation of different thesis work and science articles have been reviewed, summarized and criticized. This was the first part of the precess, since knowledge in the area is important to be able to continue with the research. Throughout the research a literature review has been pursued.

1.3.2 Modeling in MATLAB

MATLAB is known worldwide by engineers and scientists as a tool for modelling and computations. MATLAB is used to create the model, minimization, the VISSIM COM interface and for the implementation of controller. The METANET model was implemented and is a model that is commonly used to model traffic control measures e.g. VSL or ramp metering. The model is described in detail in section 2.1.

1.3.3 Orientation in simulation software and implementing COM interface

Getting to know the simulation software, VISSIM, is necessary and is performed by help from tutorials, manuals and by setting up smaller networks. VISSIM is described in chapter 4. To be able to implement speed limits in the simulation software, a COM interface needs to be created that allows the user to read and set values during a simulation. Information about how to implement the COM interface and important methods in the process was reviewed and applied. The COM interface is described in detail in section 4.4.

1.3.4 Controller design

To design a proper controller that suits the system, it is important to get to know the system and the properties of it, to know what kind of controller that is best suited for the system. It is necessary to examine different controllers to be able to tell which one that has the desired properties. After a controller was chosen, information about how to implement it was gathered and reviewed. The controller design is described in detail in chapter 6.

1.3.5 Study visit/case study

When enough information is gathered and familiarity within the area is established, a study visit was proceeded. The purpose with the study visit was to ask more specific questions to the employees at the Swedish Transport Administration who works within the area of VMS and VSL. The main purpose with the visit was to gain a deeper understanding in how the Swedish Transport Administration work with these measurements and how the situation in Gothenburg looks. During the study visit, information about the case study was received. Real time data was also collected from the selected stretch of road from the website (Vägverket, 2016).

1.3.6 Evaluation

A great number of simulations were performed and evaluated continuously. Graphs that were conducted from MATLAB had a great impact on understanding the results. Plots from simulations were compared with plots from other simulations, with different scenarios, and also compared with plots from literature to validate the results.

1.4 Outline of the report

In chapter 2, a presentation of traffic flow models and a description of macroscopic and microscopic modeling is done. This follows by a chapter that gives information about traffic control measures, e.g. VSL and VMS. That chapter also gives a picture of how VSL and VMS works in Gothenburg and in Europe today. The fourth chapter presents the software VISSIM, which is used for traffic simulations. In the same chapter, an explanation about the COM interface between MATLAB and VISSIM is also presented. In chapter 5, a calibration of the model is presented. The purpose is to minimize the difference between the model and the simulated data. Chapter 6 presents the controller design and the usage of the controller. In chapter 7, the case study is presented, together with the results from the simulations and the implemented controller. In chapter 8, a discussion of the results is done and the last chapter presents the conclusion of the research.

2 Traffic flow models

In this chapter, traffic flow models is presented. First, the macroscopic model METANET is presented, followed by the extension Cremer's models and in the last section, modeling congestion with METANET is discussed. This section will also give examples of how different congestion looks like with and without the use of already existing VSL models.

To emulate the traffic flow, mathematical models is used. This makes it possible to compare potential improvements and evaluate traffic control measures. A commonly used model is the METANET model (Messner and Papageorgiou, 1990), which can be used to model traffic control measures such as ramp metering and VSL. To be able to describe traffic processes with mathematical models, it is preferable to differentiate how detailed the description should be. During traffic modeling, a microscopic or macroscopic level of detail often is used. The macroscopic modeling is more abstract while the microscopic model resemble the real traffic flow in a more concrete way. This is explained by the following:

- **Microscopic** models describes the behavior of vehicles individually. These type of models contains movement of individual vehicles which the macroscopic model do not process. Car-following and lane-changing are significant aspects of this model. This is also the approach that usually is used during simulations (Messner and Papageorgiou, 1990).
- **Macroscopic** models describes the traffic flow as a fluid with specific variables, such as density, speed and flow. These types of models do not consider the movement of individual vehicles, such as lane-changing etc. The macroscopic description expresses the behavior of traffic at specific times and locations (Messner and Papageorgiou, 1990).

2.1 METANET

The most commonly used macroscopic model for traffic modeling is the METANET model. This macroscopic model describes traffic flow with variables such as:

- Density $\rho_{m,i}(k)$ $[veh/km/lane]$
- Mean speed $v_{m,i}$(k) $[km/h]$
- Flow $q_{m,i}(k)$ $[veh/h]$

As written above, the macroscopic description expresses the behavior of traffic at specific times and locations where time and space inputs are discretized. This results in a nonlinear analytic discrete-time model called METANET that can be used to simulate traffic flow in motorways and to test different control strategies (Papageorgiou et al., 2010). T represents the time-discrete step ($T = 10s$ is a common choice). The relation between discrete and continuous time is stated as $t = kT, k = 0, 1, 2, ..., K$ where K is the time horizon. Subscript m denotes a motorway link which is divided into $i = 1, 2, ..., N_m$ segments. All segments are of the same length, L_m, such as the criterion $L_m \geq Tv_{f,m}$, where $v_{f,m}$ is the free speed on link m, is satisfied. The motorway link has λ_m number of lanes. An example of a segment is shown in figure 2.1.

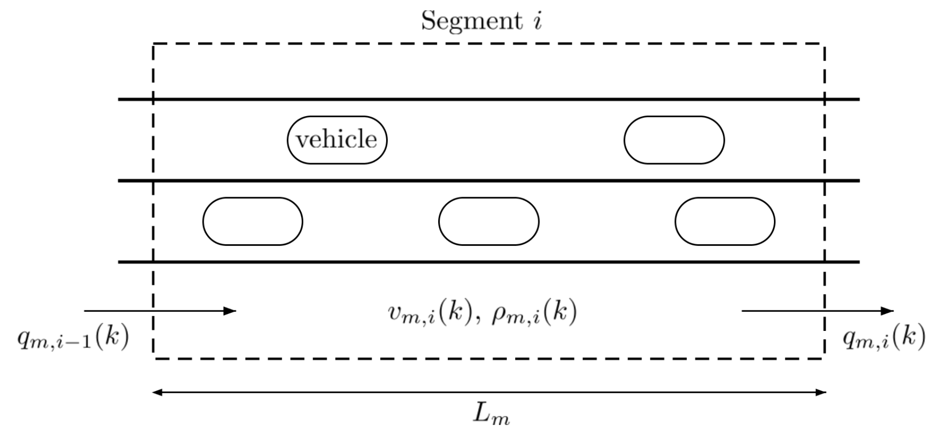


Figure 2.1: Segment i of motorway link m, with $\lambda_m = 2$

2.1.1 Link equations

Each link has the same features and there are no on-ramps or off-ramps included in the following equations. This means that the structure of the link is simple and the small changes of the structure are of no importance.

The most important equations from METANET are stated below:

$$\rho_{m,i}(k+1) = \rho_{m,i}(k) + \frac{T}{L_m \lambda_m}[q_{m,i-1}(k) - q_{m,i}(k)] \tag{2.1}$$

Equation (2.1) is the conservation equation, which is describing how the number of vehicles in the next time unit is depending on the last value of vehicles in the segment and the number of incoming and exiting vehicles.

$$q_{m,i}(k) = \rho_{m,i}(k) v_{m,1}(k) \lambda_m \tag{2.2}$$

Equation (2.2), the flow equation, describes the relation between flow, density, mean speed and number of lanes.

$$\begin{aligned} v_{m,i}(k+1) = v_{m,i}(k) &+ \frac{T}{\tau}\{V[\rho_{m,i}(k)] - v_{m,i}(k)\} \\ &+ \frac{T}{L_m}[v_{m,i-1}(k) - v_{m,i}(k)]v_{m,i}(k) - \frac{\nu T}{\tau L_m}\frac{\rho_{m,i+1}(k) - \rho_{m,i}(k)}{\rho_{m,i}(k) + \kappa} \end{aligned} \tag{2.3}$$

Equation (2.3) is the empirical dynamic speed equation, where $v_{m,i}(k)$ (first term) explains that the value of the mean speed at the next time unit depends on the previous mean speed. The second term expresses that drivers tries to reach equilibrium speed. The third term describes how the drivers adjusts their speed to new conditions in the entered segment. If the entered segment is more crowded than the previous segment, the driver will probably decrease the speed. The last term describes how the drivers adjust their current speed to what they anticipate will happen by reason of what they see in the nextcoming segment. The time constant τ, the anticipation constant ν and κ are design parameters that are equivalent for each link and can be determined and optimized with a validation process.

$$V[\rho_{m,i}(k)] = v_{f,m} exp\left[-\frac{1}{\alpha_m}\left(\frac{\rho_{m,i}(k)}{\rho_{cr,m}}\right)^{\alpha_m}\right] \tag{2.4}$$

In equation (2.4), V describes the relations between mean speed and density. ρ_{cr} defines the critical density and α_m is a design parameter (Papageorgiou et al., 2010). V can also be used to plot a fundamental diagram and to calculate the mean speed.

2.1.2 The fundamental diagram

To be able to see that the traffic behavior is realistic during the congestion phase, the flow can be plotted against the density to get a fundamental diagram, see figure 2.2. Down in the left hand corner of the diagram, the density and flow is low, which allows drivers to travel close to the maximum allowed speed on the road. The relationship between the density and the flow is at this point almost linear. Increasing flow and density leads to a more and more congested road, this also leads to a decreased speed from the drivers. When the critical density is reached, the flow is decreasing since the drivers speed is also decreasing. The critical speed is the speed reached when the flow capacity is attained (Hegyi, 2004). If the plotted flow versus density curve looks approximately like figure 2.2, the model is realistic.

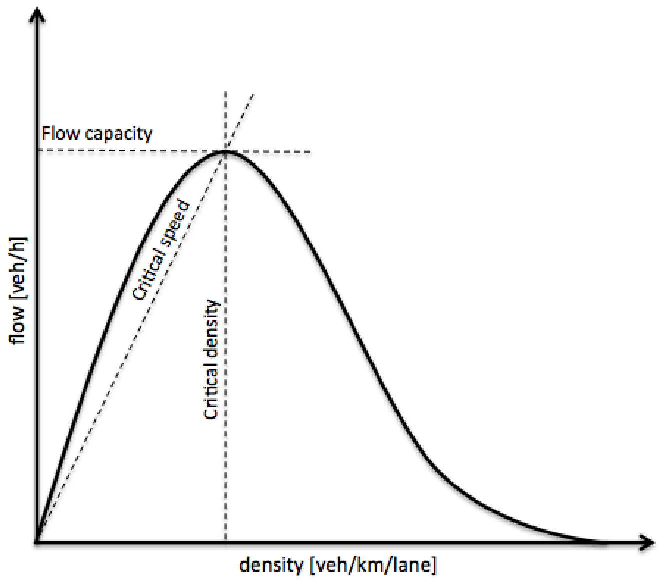


Figure 2.2: An example of a fundamental diagram

2.1.3 On-ramp equations

The characteristics for a certain link depends on both main-stream on the motorway, but also on the flow entering the link from on-ramps.

$$w_o(k+1) = w_o(k) + T(d_o(k) - q_o(k)) \tag{2.5}$$

Equation (2.5) is used when to combine on- or off-ramps with the desired link. $w_o(k+1)$ is the queue length at origin o in the next time interval which depends on the queue length in the current time interval ($w_o(k)$). $d_o(k)$ is the demand flow and $q_o(k)$ is the outflow from the ramp. This is expressed in figure 2.3 below.

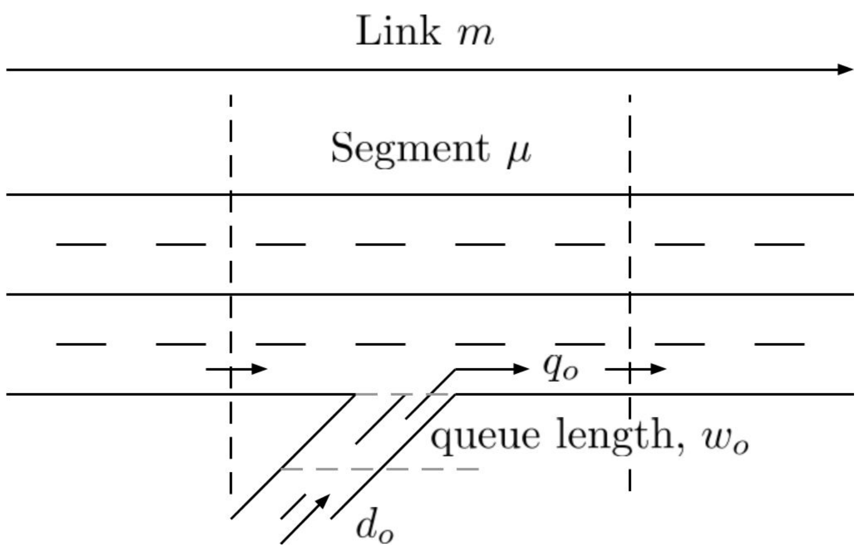


Figure 2.3: Traffic conditions for a segment where an on-ramp is connected depends on the characteristics for the ramp and the segment. The queue length depends on the amount of vehicles that wants to enter the ramp and the amount of cars that is entering the segment and leaving the ramp.

The following four equations describes the outflow from the on-ramp:

$$q_o(k) = r_o(k)\hat{q}_o(k) \tag{2.6}$$

$$\hat{q}_o(k) = min\{\hat{q}_{o,1}(k), \hat{q}_{o,2}(k)\} \tag{2.7}$$

$$\hat{q}_{o,1}(k) = d_0(k) + \frac{w_o(k)}{T} \tag{2.8}$$

$$\hat{q}_{o,2}(k) = Q_o \; min\left\{1, \frac{\rho_{max} - \rho_{\mu,1}(k)}{\rho_{max} - \rho_{cr,\mu}(k)}\right\} \tag{2.9}$$

r_o is the ramp metering rate, if no ramp metering is applied $r_o = 1$. Q_o is the ramp's flow capacity, ρ_{max} is the maximum density for the link and $\rho_{\mu,1}$ is the density for the segment that the on-ramp is connected to (Papageorgiou et al., 2010).

$$-\frac{\delta T q_o(k) v_{m,\mu}(k)}{L_m \lambda_m (\rho_{m,\mu}(k) + \kappa)} \tag{2.10}$$

Term (2.10) is describing the speed drop caused by alternative on-ramps. This term is added to equation (2.3) for segment μ where the on-ramp are connected. δ and κ are design parameters describing the appearance of the road. $q_o(k)$ is the ramp flow from origin o and $v_{m,\mu}(k)$ and $\rho_{m,\mu}(k)$ are the speed and density of the segment connected to the ramp. (Hegyi, 2004)

2.1.4 Boundary conditions

Boundary conditions needs to be specified for segments located at the entry and exit of the network. The flow and the mean speed in the current segment depends on the previous segment, therefore those variables needs to be defined for the entry of the network. The mean speed in current segment also depends on the density in the next segment, which means that the density needs to be defined for the last segment of the network (Hegyi, 2004).

The values for the boundary conditions depends on what kind of situation the model should emulate. It can either be a default value or a time dependent function, e.g. a congestion can be modeled by specifying a high flow in the entering segment. This means that there are many cars entering the network which will give a congestion that propagate forward in the network. A congestion can also be emulated by selecting a high density in the last segment. This will, unlike the previous example, give a congestion propagating backwards in the link.

2.2 Cremer's models

To implement VSL in the METANET model, two models suggested by Cremer can be used. Equation (2.4) is replaced by the models for VSL in METANET and a control varaible is used to control the speed. The first model for modeling VSL is (model 1):

$$V(\rho, u) = v_{f,m} u \exp\left(-\frac{(1+u)}{4}\left(\frac{\rho}{\rho_{cr}}\right)^2\right) \tag{2.11}$$

and the second by (model 2):

$$V(\rho, u) = v_{f,m} u \left(1 - \left(\frac{\rho}{\rho_{max}}\right)^{(l-1)(3-2u)}\right)^{\frac{1}{1-m}} \tag{2.12}$$

Where $v_{f,m}$ is the free speed on link m, ρ_{cr} is the critical density and ρ_{max} is the congestion density. The input that controls the speed are u and the parameters l and m are fitting parameters. There is a specific relation between the control variable u and the speed limit depending on the free flow speed given for the network (Hegyi, 2004).

Both model 1 and model 2 have some disadvantages. A problem with model 1 is that the mean speed can sometimes be unrealistically low. A study has shown that if the speed limit at free flow on a certain road is 100 km/h and the density increases so the speed limit decreases to 75 km/h the actual speed was 60 km/h. This scenario is unrealistic because if 100 km/h is the normal speed limit, drivers will probably not drive in 60 km/h if they are allowed to drive in 75 km/h. Another problem with model 1 is that the speed-density and the flow-density characteristics are approximately scaled with the control variable u. This means that even though the density is not high enough to result in controlled speed limits, the speed will still be decreased (Hegyi, 2004).

The weakness of model 2 is the behavior of the equilibrium speed at medium density (around critical density) and at high density (around maximum density). When the speed limits are decreased, due to increased density, the equilibrium speed is increased. The resulting behavior can be explained by higher variance of the speed of individual vehicles in the case when the speed is controlled compared to the case with no control of speed limits. However, this does not implicate that the behavior of the equilibrium speed is unrealistic because a smoother traffic flow results in higher speed at a specific density which also leads to higher traffic flow, nevertheless field studies have shown that this is not the case (Hegyi, 2004).

2.3 Modeling congestion

To be able to model the traffic flow and congestion with METANET, the equations can be implemented in MATLAB. Equation (2.1)-(2.6) can be coded and the variables mean speed, density, flow and equilibrium speed with and without VSL are implemented as matrices. The rows represents the segments of the link and the columns represents every time unit. The elements in the matrices are calculated with help of two nestled for-loops. The code can be found in Appendix A.

In section 2.2 it is described how to model congestion with METANET in two different ways with help of boundary conditions. To visualize this, two scenarios are implemented in MATLAB. The first scenario has a rush in flow to the entering segment between the time t=80 and t=130 in combination with lower speed for the vehicles. This gives a congestion that will propagate forward in the network. The second scenario has a heightened value for the density in the exiting segment between the time t=50 and t=100. This means that the density is dense and traffic will be congested at this time. The traffic jam for this setting will propagate backward in the network. The following figures shows the results for the two scenarios for each of the two models which are; without VSL, with VSL model 1 and with VSL model 2.

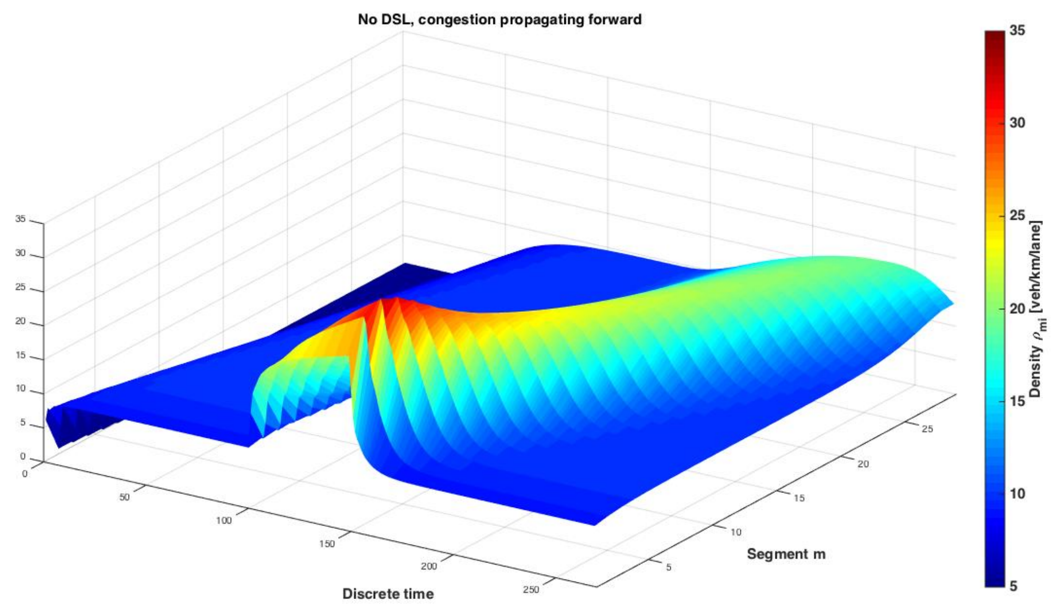


Figure 2.4: Congestion propagating forward, without VSL

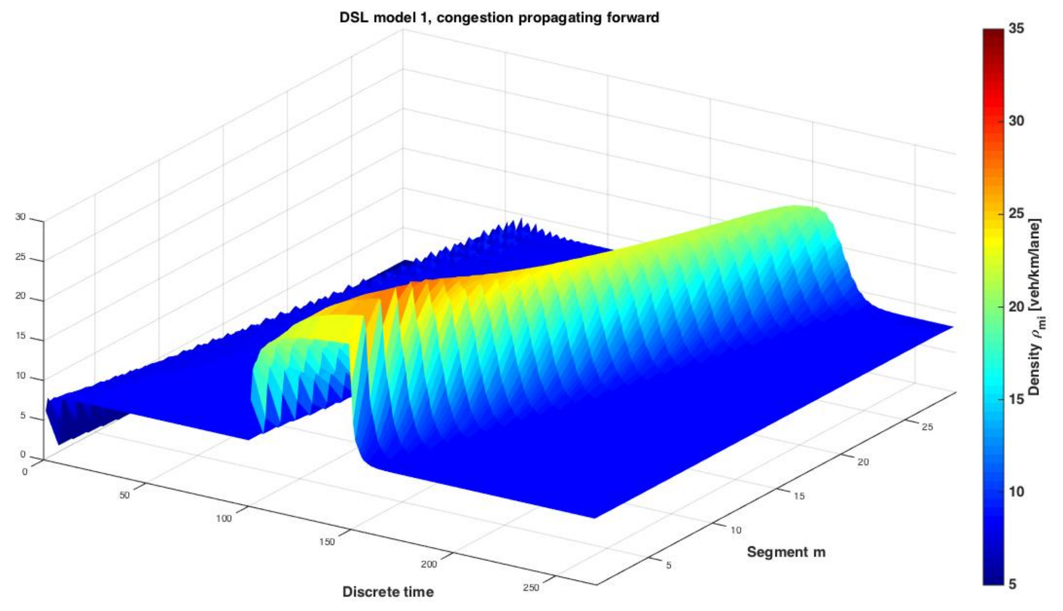


Figure 2.5: Congestion propagating forward, with VSL model 1

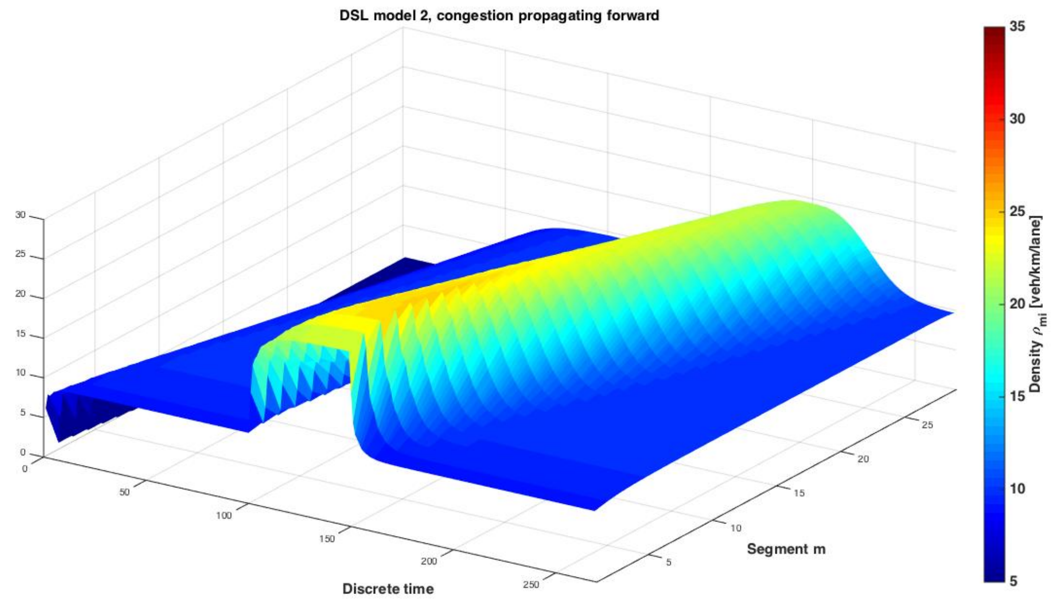


Figure 2.6: Congestion propagating forward, with VSL model 2

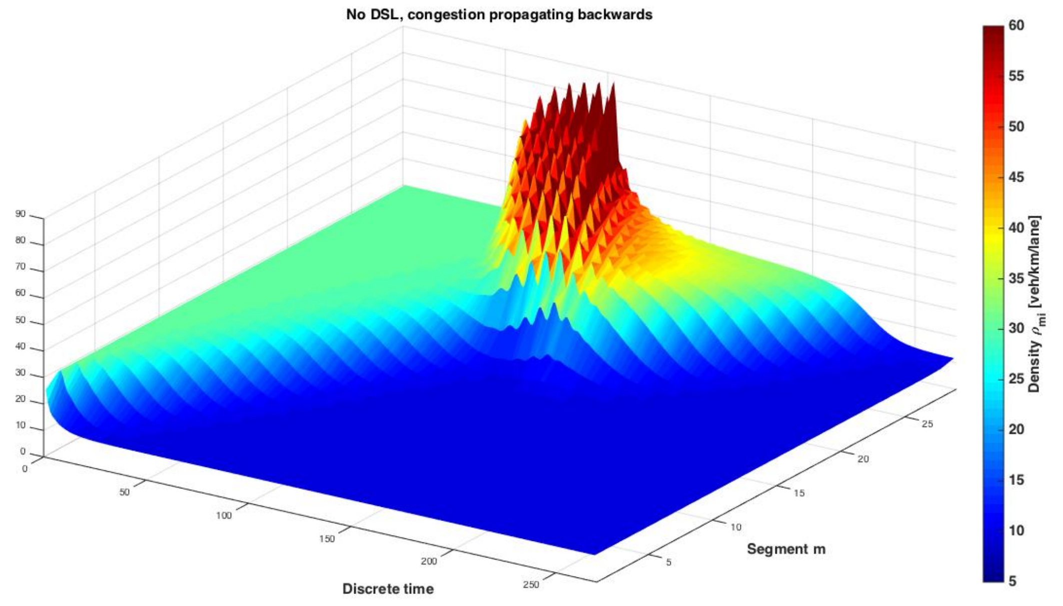


Figure 2.7: Congestion propagating backwards, without VSL

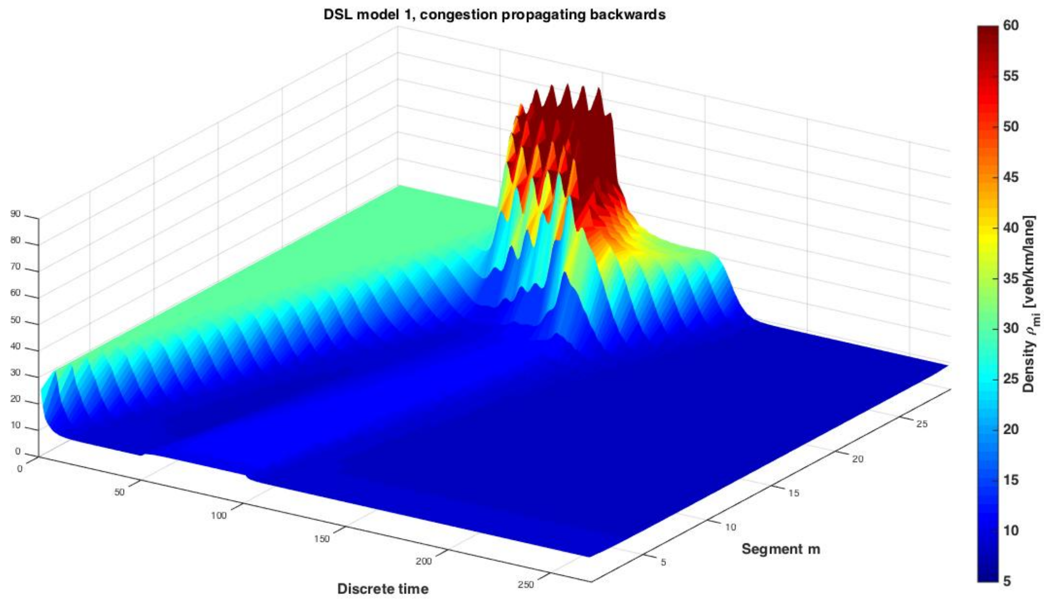


Figure 2.8: Congestion propagating backwards, with VSL model 1

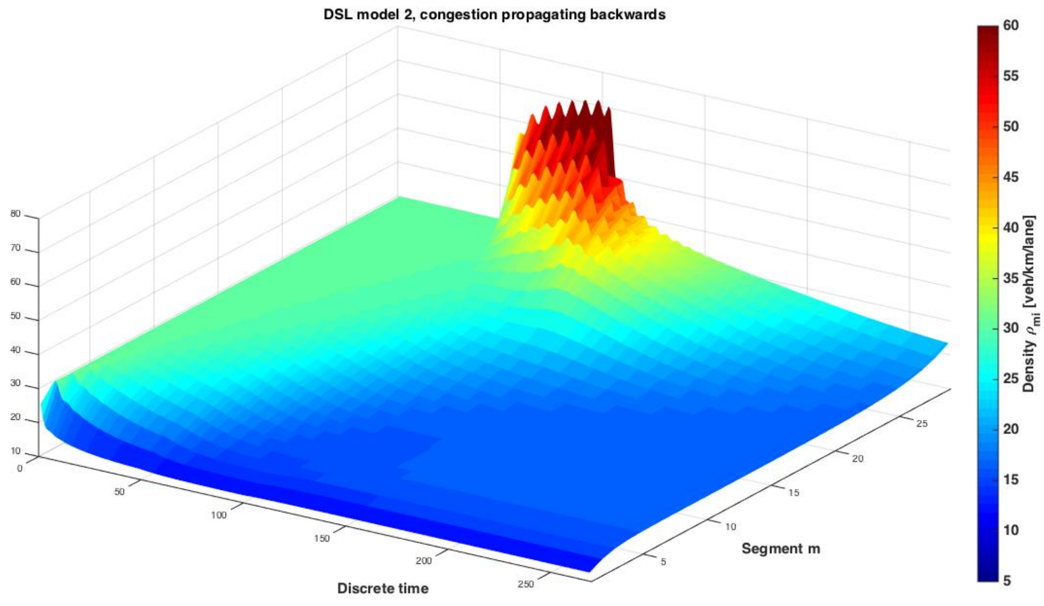


Figure 2.9: Congestion propagating backwards, with VSL model 2

The colours in the graphs specifies the density level. Red and yellow indicates higher density and blue indicates lower density. The congestion in each picture is indicated by the red and yellow areas. The first three pictures emulates a scenario where the congestion propagates forward in the link. Figure 2.4 shows the case where no VSL is applied and it is clear that the yellow shade is stronger than when a VSL model is implemented. The first model for VSL, figure 2.5, shows improvements since the density does not get as high as in the uncontrolled case. Though, the improvement is not as distinct as in the previous scenario where the congestion propagates backwards. In figure 2.6, where the second model for VSL is implemented, improvement are noticed in the density. The congested area is not as high as in the uncontrolled case and the congestion also fades away faster than in the uncontrolled case.

The last three pictures represents the case where the congestion is propagating backwards in the link. In figure 2.7 no control for VSL is used and it is clear that this scenario has the highest level of congestion and improvements can be made. Figure 2.8 and 2.9 has model (1) and (2) for VSL implemented and both clearly indicates improvements. The first VSL model, figure 2.8, makes the congestion fade faster which is shown by the reduced red and yellow area. When the congestion has faded, the density gets low and the characteristics for the link is the same as in the uncontrolled scenario. When the second model for VSL is implemented, figure 2.9, the congestion also fades faster and the density in the congestion phase in the first segment does not get as high as in the uncontrolled scenario or in the figure 2.8 where model 1 for VSL is implemented. This is followed by a higher density for the whole link, but due to that the density does not exceed/is close to the critical density this is not a problem. When the density exceeds the critical density the link gets congested i.e. a yellow area becomes visible.

3 Traffic control measures

There are different measures to control and improve the performance in a traffic network. In sections 3.1-3.5 some of the most common used traffic control measures (TCM) will be presented and explained. The main focus will be on VMS and VSL and the combination of the two, but other measures will also be briefly presented. In section 3.6-3.7 the current use of VMS and VSL in Gothenburg and in Europe will be presented.

3.1 Variable message signs

Variable message signs is electronic signs mainly used on, or in connection to freeways. The signs are being used to display different types of messages to drivers. These messages may be changed and switched on or off depending on the current traffic situation and the requirements of the road. The signs displays messages in terms of pictograms, text or a combination of both (Ronchi et al., 2016). An example is overhead gantry signs in tunnel management that shows available lanes to enter the tunnel. Speed limit signs that changes depending on the conditions in the traffic or the weather conditions is another example. VMS is also used for displaying the level of occupancy in a car park, messages triggered by speeders (Nygårdhs and Helmers, 2007) and estimated travel times (Abdel-Aty and Yu, 2014).

Route guidance is another measure being used to prevent the occurrence of traffic congestion. It provides information to help the driver choose a better alternative route. The information is typically being showed on variable message signs. This information can be travel times to different destinations, queue times or delays. The driver can then choose the best route for his destination taking the given information into consideration (Hegyi, 2004).

When displaying VMS it is important to consider the driver expectations and the credibility of the information. VMS signs should only show messages of importance or relevance to the drivers, otherwise the messages will not draw attention or be complied by the drivers (Nygårdhs and Helmers, 2007).

3.2 Variable speed limits

Variable speed limits, also known as dynamic speed limits, is a control strategy that helps or warn drivers to adjust their speed when the traffic conditions are changing. The speed limits can be either mandatory or recommended, where the mandatory signs has a red circle around the speed limit (Nissan and Koutsopoulosb, 2011). VSL has several different objectives. The most common objectives are: to reduce recurrent congestion, improve traffic safety, improve traffic during adverse weather conditions and to decrease emissions (Abdel-Aty and Yu, 2014). Reducing recurrent congestion is the main objective when implementing VSL on freeways today.

There are two approaches when aiming to reduce recurrent congestion: the homogenization approach and the traffic breakdown prevention approach. The homogenization approach aims to go above the critical speed when choosing speed limits (Hegyi et al., 2005). A more homogeneous traffic flow is a typical result of reducing recurrent congestion, and further leads to a safer traffic environment (Hegyi, 2004). Another positive effect from a homogeneous traffic flow is improved air quality. However, this approach does not improve traffic volume because the speed limits does not limit the traffic flow. The traffic breakdown prevention approach aims to go below the critical speed when choosing speed limits, in contrast to the homogenization approach. The traffic breakdown prevention approach aims to prevent too high densities and limit the inflow (Hegyi et al., 2005).

In addition to these two approaches other interesting applications can be the use of VSL on links with sharp curves, then aiming to prevent precipitous breaking. VSL can also be applied in merging or weaving areas to coordinate the speed of the merging or weaving vehicles to produce a more smooth traffic flow (Hegyi, 2004).

The most common way to evaluate the VSL system is to look at traffic flow parameters such as average speed, travel time and traffic volume before and after VSL is applied to a motorway system. There is also a statistic way to measure the capacity of the motorway. The basic idea of the statistic way is to establish two models based on density by using the average of speed data and traffic flow of 5 minutes intervals. Equalities and coefficients for the models will then be checked for by using a generalized F-test (Abdel-Aty and Yu, 2014).

3.3 Combining VMS and VSL

The location of the displaying devices for VMS and VSL is either road-side signs or overhead gantry signs, see figure 3.1, but the overhead gantry signs is the ones most commonly used. To provide more information to drivers, a combination of VMS and VSL is often used. This combination provides the driver with a new speed limit and the reason why the speed limit was changed (Abdel-Aty and Yu, 2014). This combination with a VSL in addition with the explanatory information has been concluded to be more effective and show a higher compliance level from the drivers, in comparison with the use of VSL solely (Nygårdhs and Helmers, 2007).

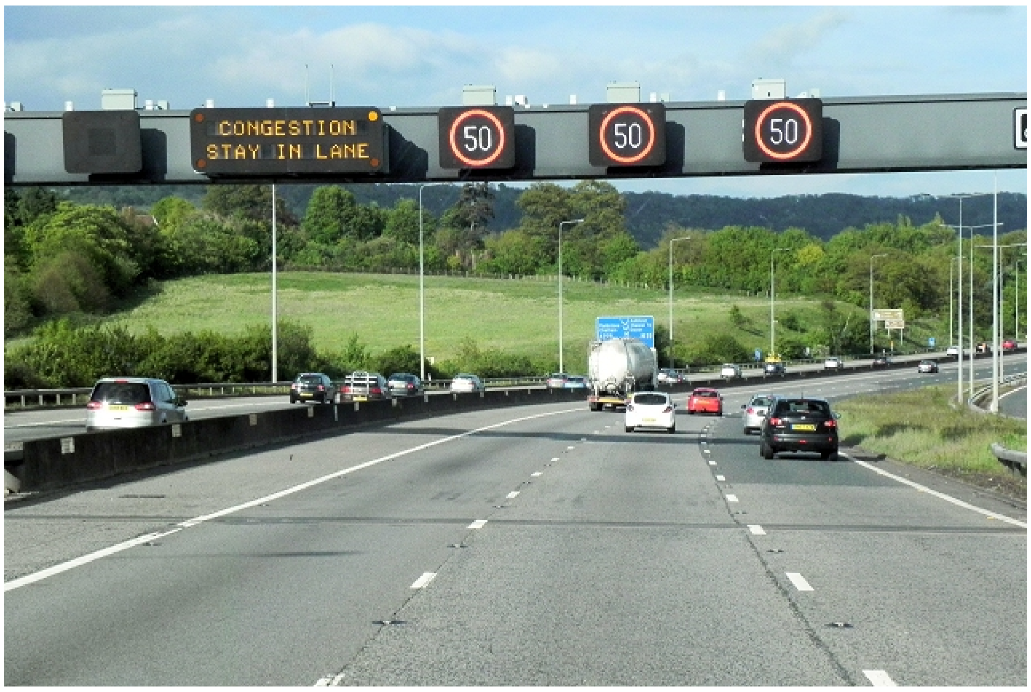


Figure 3.1: Overhead gantry sign combining VMS and VSL (David Dixon, 2013)

3.4 Other control measures

Ramp metering is a well studded traffic control measure that determines the entering of traffic flow on the freeway. A traffic light regulates the flow on the on-ramp and the light changing intervals decides the flow rate. There are two different modes when using ramp metering. The first is traffic spreading mode that takes the mean value of the incoming vehicles and sets this value as the metering rate. The purpose of this mode is to spread the vehicles and is useful when vehicles arrives in group e.g. from another controlled area. Traffic restricting is the other ramp metering mode. This method is used when drivers try to avoid upcoming congestion by taking another route through a local road or when the traffic is dense. The restrictive mode aims to prevent traffic break down by regulating the incoming traffic flow rate in order to keep the density on the freeway below the critical density (Hegyi, 2004).

3.5 Coordination and design of VMS controller

There are three different existing techniques for coordinating a VMS controller: knowledge-based methods, model-based optimal control methods and control parameter optimization methods (Hegyi, 2004).

The knowledge-based traffic control methods describes a combination of the control system, in terms that are understandable for humans, and knowledge about the traffic system. The aim is to assist operators in traffic control centers to find a good, but not necessarily the best, combination of control measures and a suitable solution due to the current traffic situation (Hegyi, 2004).

The approach for model-based optimal traffic control predicts the traffic systems future behavior by using a model based on the existing traffic condition, the traffic demand expected on the network level and the traffic control measures planned. By selecting a scenario for the traffic control measures, an optimization of the traffic systems future performance is possible (Hegyi, 2004).

The control parameter optimization methods uses a control law for ramp metering and speed limit control. By optimizing the mean performance through simulation of many scenarios of traffic situations, improved parameters for the control law can be found and used (Hegyi, 2004).

3.5.1 Model Predictive Control

To design the VMS controller an approach called model predictive control (MPC) can be used. The approach is a model based optimal control method and predicts the future traffic behavior and determines optimal speed limits by using a macroscopic traffic model (Nissan and Koutsopoulosb, 2011). The method is integrating multivariable control, optimal control and processes in control with dead times. MPC also uses future references when available (Bordons and Camacho, 1998).

The basic principles of MPC is to predict the future behavior of the system. A certain time-horizon will be used to predict the systems future behavior. The prediction is based on: the control signal planned, the systems current state and the disturbances expected in the system. After that an evaluation of the performance is done, due to a user-specified objective function that is based on planned control signal and the systems outputs and states. The objective function is later optimized with a control signal. The last step is the predictive control that uses a rolling horizon scheme to recalculate the control signal. The control signals first samples is added to the process and the remaining parts are recalculated for every sample step of the controller (Hegyi, 2004).

3.6 Traffic control in Gothenburg

The Swedish Transport Administration in Gothenburg use a system for control of the motorway called Motorway Control System (MCS). This system is a collecting name for various system solutions. It is used all around the world for increasing the efficiency of the road network. The Swedish Transport Administration uses a system called Motorway Traffic Management System, MTM_2, and this system is based on the international standard and is a type of system solution. MTM_2 is developed in collaboration with the Dutch National Road Administration and the system has been in use for over 30 years. This system is based on microwave detectors placed above each lane at every portal. Data for the flow and speed is collected by the microwave detectors, then the system analyzes the traffic data to estimate if a lower recommended speed should be applied, to warn drivers about a possible traffic congestion.

Besides the automatic local function that is used to collect data, there is also a central unit that help to convey information between the different portals. With the help of the central system the traffic controllers can carry out manual actions e.g. close a traffic lane due to a traffic accident or road construction. When it is required, the recommended speed limits can be changed to mandatory (red circle around the speed limit) by manually changing it, figure 3.2[2] shows this.

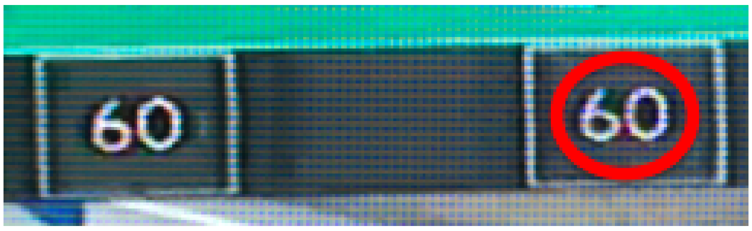


Figure 3.2: Recommended VSL and mandatory VSL

Today the speed of the vehicles is the only parameter that is in focus when finding out how the traffic situation has changed and if it is necessary to warn the drivers by changing the speed limits on the signs. The reason for only considering the speed is that The Swedish Transport Administration uses an old system and that is difficult to change with respect to algorithms and queue detections.

VMS and VSL are today implemented in a few different areas in Gothenburg, e.g. in the area "Åbromotet Norra" see figure 3.3[2]. According to Pernilla Fransson[1], an expand of VMS and VSL is in progress. The development of the system MTM_2 that is used today has terminated, also the supply needed for the MTM_2 system is expensive and requires a lot of maintenance. Instead, the Swedish Transport Administration is running a project were they are working with developing a new system called, LASSY, LAne Signaling SYstem. The vision according to the Swedish Transport Administration with LASSY is that they can get more cost-effective solutions where the number of equipment along the road is minimized.

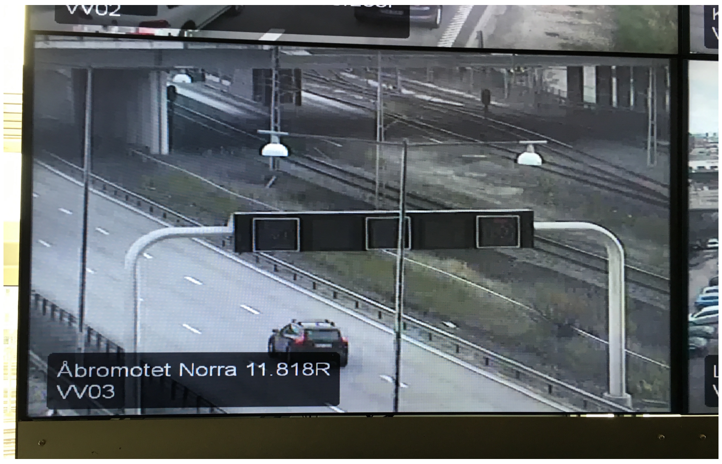


Figure 3.3: TV-screen at the Swedish Transport Administration that shows a VSL sign in Gothenburg

Today the Swedish Transport Administration uses a network software called National Traffic System (NTS) for controlling the signs. As Figure 3.4[2] shows, how the traffic is controlled with the help of multiple screens that shows how the traffic looks by cameras that are placed on gantries. They can get calls from civilians or the police about an incident and can then easily

[1]Pernilla Fransson is the branch head of traffic information and traffic control for traffic control region west.
[2]Picture taken during the study visit at the Swedish Transport Administration.

switch to a camera in that area and analyze the situation and adapt the VMS and VSL to the situation.

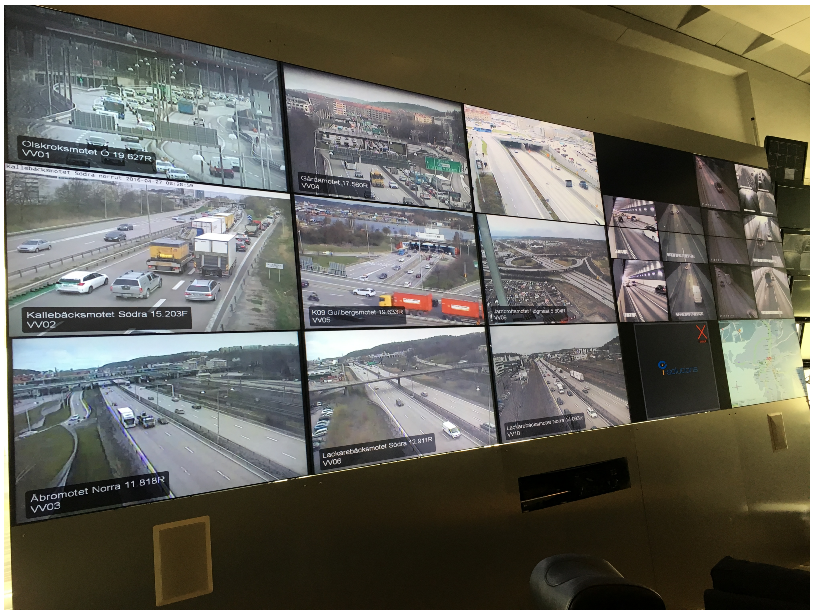


Figure 3.4: Control room at the Swedish Transport Administration

There is a website called Trafiken.nu that has information about the traffic situation in Sweden. On the website, it is possible to select a specific area or road to see the current traffic situation, e.g. how congested a road is at the moment. It is also possible to use it to plan a trip by looking at which periods of the day the road is congested. Figure 3.5 shows how the website looks and that the roads change from green, yellow and red depending on how congested the roads are.

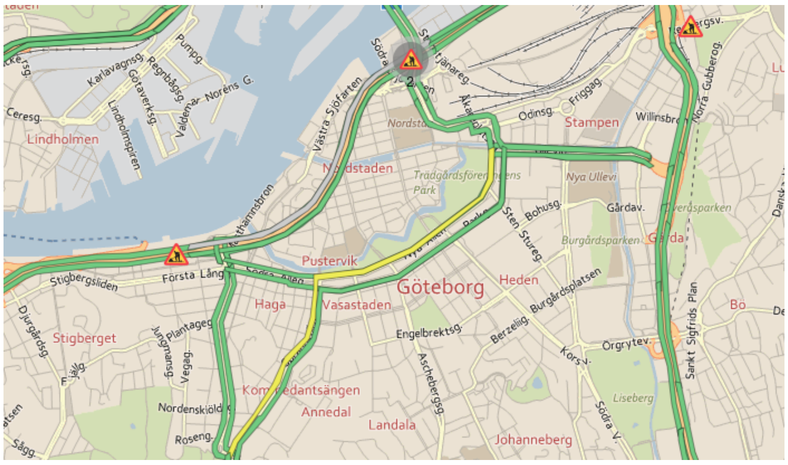


Figure 3.5: A screenshot from Trafiken.nu

The Swedish Transport Administration consider the opportunities to collect traffic data from different systems which allows various types of algorithms to be used to get a better reliability and prognosis. Eventually they want to make it possible to distribute data between different system but today that is hard. Also the Swedish Transport Administration expresses that they would like the possibility to optimize their algorithms so they are adapted based on the current road section, time of day or by season. A system that reduces the speed automatically in some sections based on the weather and/or the level of particles (emissions from cars) is requested, since it has to be changed manually by traffic controllers in the system that is used today[3].

3.7 Use of VMS and VSL in Europe today

Today there are several countries in Europe that use VSL or VMS, some of these are England, the Netherlands, Germany, Finland and Sweden. Different countries consider different parameters when displaying speed limits and messages. For the motorways in United Kingdom they employ dynamic and simple matrices control algorithm. It works so when the volume reaches a specific level (vehicles per hour per lane) the speed limit is reduced (Abdel-Aty and Yu, 2014).

In the Netherlands on the A2 (Autobahn) the given speed limits are established by a control algorithm based on 1-minute average speed and volume across all lanes. On A16, also in the Netherlands, the displayed speed limits is predicted on the visibility conditions gathered by 20 visibility sensors along the road. The control algorithm works so that if the visibility falls below 140 meters the speed limit will then fall to 80 km/h. If the visibility instead falls below 70 meters the speed limit will be dropped to 60 km/h (Abdel-Aty and Yu, 2014).

Meanwhile in Finland, the road conditions is the only factor that the VSL algorithm depends on. The local weather and road surface (dry, salted, wet and snowy) conditions determines the road condition, and elements like wind velocity, air temperature, rain intensity plays a major role. If the rood conditions are good the speed limits will display as 120 km/h and for bad road conditions 80 km/h(Abdel-Aty and Yu, 2014).

The speed limits used on German motorways depends on volume, speed and environmental data (fog, ice, wind and other detectors). To predict the conditions over the coming 30 minutes historic data is used in addition with the volume of passenger cars and trucks speed (Abdel-Aty and Yu, 2014).

[3]Information about traffic control in Gothenburg was recieved from Emil Muda working at infrateknik Underhållsdistrikt Väst.

4 Traffic simulation in VISSIM

In this chapter, the simulation software VISSIM is introduced and a short guide is presented. Section 4.1 describes how to implement a network in VISSIM, followed by information on how to do a simulation and evaluate the data. The last section describes the VISSIM COM interface, how to implement it and how to use it in combination with MATLAB.

Traffic simulations can be a useful tool in producing an authentic model. They can strengthen and improve the model. Traffic simulations can also be done in order to research the effects of implementing VSL into traffic networks as well as other measures. The traffic simulations in this research were made with PTV VISSIM software. VISSIM is a software especially developed for traffic simulations. The software is used to simulate microscopic models where each vehicle is simulated individually. Results can be collected and evaluated either from nodes or from links and their segments. Results that can be collected are for example flow, density and speed.

4.1 Network objects in VISSIM

When creating an infrastructure network in VISSIM the first step is to create links. This can be done either on a background map or on an empty workspace. The link setup tool allows the user to drag out the link and then setup the data e.g number of lanes, aesthetics, direction. To modify the shape of the link, extra points can be added and then a part of the link can be moved in a slightly different direction. To form a curve the spline tool is useful, the more points in the spline the smoother the curve becomes. To obtain a complete network the links can be merged together. It is not enough to just overlap the links, they must be connected with a so called connectors. If the network that is being built is a motorway, on- and off-ramps can be made by creating smaller links. Where two or more links are connected or where they intersect, route priority can be added between two routes. There are two options, either the roads has equal priority or one route has higher priority than the other. If it is more likely that vehicles takes a specific route, priorities makes the simulation more realistic. Figure 4.1 shows a guide of how to change network objects in VISSIM.

After the structure of the road is created, the next step is to add speed limits on the desired links. These limits can be set up by using the tool Desired Speed Decision. With this tool speed limits can be set up with time intervals which allows different speed limits that varies over time. To generate traffic in the network the feature vehicle inputs can be used. Inputs can be set up on the desired links. When setting up the vehicle inputs parameters such as volume, vehicle type, time interval can be decided. A rush might be represented by increased volume, decreased speed limits or a combination of booth during an interval.

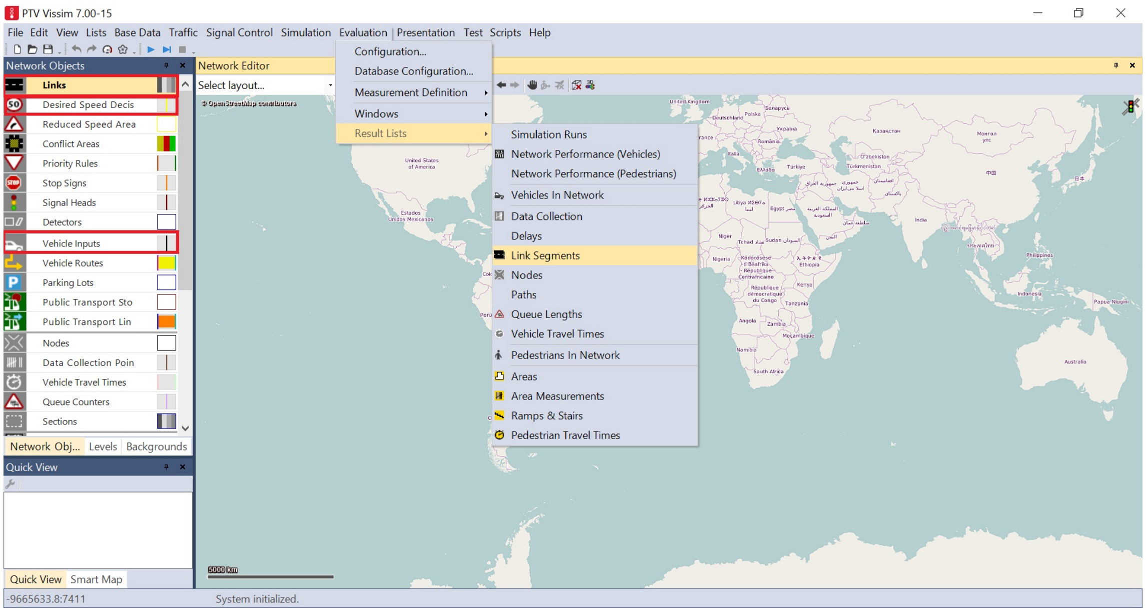


Figure 4.1: How to change network objects in VISSIM

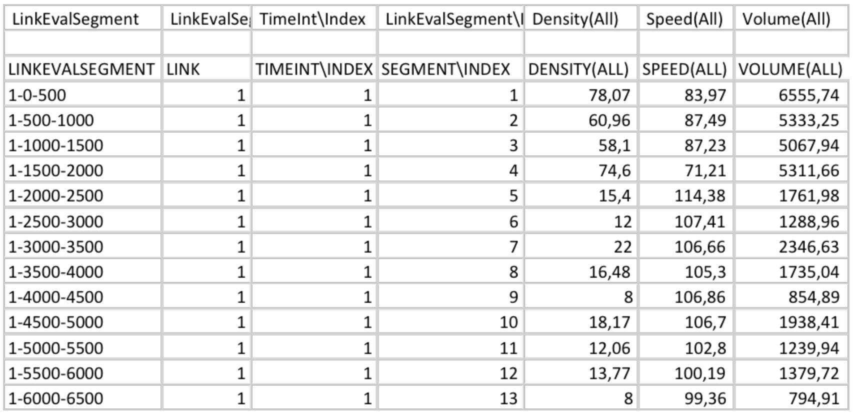
4.2 Simulation

When a network is generated and the parameters are set, a simulation may be executed, see figure 4.2. Either a simulation time can be set to a desired value or the simulation can be broken manually by clicking the stop button. In the Link Segment Results tool desired attributes can be added to the simulation output. Examples of useful attributes are the density, speed and volume in the different segments during specified time intervals. When the simulation is done the results can be saved into a separate file. To create a congestion in VISSIM there are multiple ways to go. One way is to reduce the number of traffic lanes in the last segment of the road which decreases the number of vehicles that can leave the segment and a congestion is created. Another way is to lower the speed in the last segment to around 30-50 km/h and this increases the time that the vehicles stay in that segment which causes a congestion backwards. A third alternative is to increase the vehicle input in VISSIM, since the volume of vehicles is increased the risk of congestion increases. Another way to make a congestion is to increase the traffic flow entering the road from the on-ramp.

Figure 4.2: 3D simulation in PTV VISSIM software

4.3 Evaluate simulation data

Since the network is completely empty when the simulation starts it will take a while before there are vehicles in the whole network. To make the simulation result more realistic the user can define when to start recording the result. When completing a simulation the results can be evaluated in VISSIM or saved to an external file, see figure 4.3. In the result list attributes which are of interest can be selected and displayed. There is a wide range of different results that can be evaluated e.g. density speed or volume. Results from VISSIM simulations can be saved into excel files. To connect and validate the model the VISSIM data can be uploaded into METANET by a MATLAB script. After some matrix manipulations the mean speed, density and traffic flow can be saved in global variables which makes them accessible from other scripts.



LinkEvalSegment	LinkEvalSe	TimeInt\Index	LinkEvalSegment\I	Density(All)	Speed(All)	Volume(All)
LINKEVALSEGMENT	LINK	TIMEINT\INDEX	SEGMENT\INDEX	DENSITY(ALL)	SPEED(ALL)	VOLUME(ALL)
1-0-500	1	1	1	78,07	83,97	6555,74
1-500-1000	1	1	2	60,96	87,49	5333,25
1-1000-1500	1	1	3	58,1	87,23	5067,94
1-1500-2000	1	1	4	74,6	71,21	5311,66
1-2000-2500	1	1	5	15,4	114,38	1761,98
1-2500-3000	1	1	6	12	107,41	1288,96
1-3000-3500	1	1	7	22	106,66	2346,63
1-3500-4000	1	1	8	16,48	105,3	1735,04
1-4000-4500	1	1	9	8	106,86	854,89
1-4500-5000	1	1	10	18,17	106,7	1938,41
1-5000-5500	1	1	11	12,06	102,8	1239,94
1-5500-6000	1	1	12	13,77	100,19	1379,72
1-6000-6500	1	1	13	8	99,36	794,91

Figure 4.3: Simulation results evaluated in an excel file.

4.4 VISSIM COM Interface

The VISSIM software can be used in combination with MATLAB, Java, C++ or Phyton. This is done through a COM interface which allows a wide range of additional features in combination with the existing tools in PTV VISSIM. Through the COM Interface inputs such as links, vehicle inputs, signal controllers, speed limits can be set up and controlled. When using the COM interface the simulation is being executed through the chosen programming language portal. In this research the focus will be on combining VISSIM with MATLAB through the COM interface with the main reason to create VSL controller design.

When using the COM interface in MATLAB a VISSIM object is created by the actxserver which allows access to the COM interface of VISSIM 7.00. By using different commands from the interface a project and layout can be loaded, parameters can be set, simulations can be run and results can be collected. Creating a server, loading network and loading layout are basic operations which are the first steps to complete before using the COM interface. The COM objects are hierarchically structured, meaning the object consist of fields and in the fields there are more objects with their own fields and so on. To make the MATLAB code more structured a tip is to save objects and subobjects into new variables. This coding structure also makes it easier to access the desired object. When the desired objects are created, various simulation scenarios are possible by using coding logic. A simulation can be run by setting up a time step and period or the simulation can be run step by step through a for loop in MATLAB with the "runsinglestep" command. The "runsinglestep" command is preferable in this case because of its compatibility to use collected data in algorithms and then dynamically set parameters between the executed single steps. (Bansal, 2013).

When executing a simulation different types of data can be collected or set through the objects in the COM interface. One object available in the COM interface is data collection measurements that can be used to collect traffic information such as mean speed v, or number of vehicles veh/h. These attributes can then be used to calculate other useful parameters e.g. flow q and density ρ. To reach an object through the COM interface the object must already be created in the running VISSIM file. Data collection points can be set up on optional locations on the road in VISSIM and are auto numbered by a key. It is from these points that the measurements can be extracted. Here is a code example on how to access data collection measurements in MATLAB: $datapoints = vnet.DataCollectionMeasurements$. To access a specific data point the command: datapoints$\{i\}$ = datapoints.ItemByKey(i) can be used, where the datapoints are being stored in a cell array where i is a specific number answering to the key of the desired object. To get the measurements from the desired datapoint, i, the command: datapoints$\{i\}$.get('AttValue','Speed(Current, Last, All)'); gives the mean speed from the current simulation, in the last completed time step, for all vehicle classes. Meanwhile the line: datapoints$\{i\}$.get('AttValue','Vehs(Current, Last, All)'); gives the vehicles passing the datapoint in the current simulation run during the last completed time step and for all vehicle classes. When Accessing another type of object in the COM interface the same logic as mentioned above is applied but the handle "$DataCollectionMeasurments$" is being replaced with the desired objects handle. The steps above can then be done e.g. for setting up a speed limit with the commands:

speeddecision = vnet.DesSpeedDecisions;
speeddecision_1 = speeddecision.ItemByKey(1);
speeddecision_1.set('AttValue', 'DesSpeedDistr(10)', 100);

Where "$DesSpeedDistr(10)$" indicates which vehicle types the speed decision should apply to and 100 is the desired speed limit. Note that speed decisions only can be set to integer values and also make sure that the desired speed limit is available in VISSIM, otherwise there will be a runtime error. With the tools presented above VSL can be implemented in VISSIM. The results can then be analyzed by saving the results into excel files and plotting in MATLAB or just observed directly in the software.

5 Model calibration

In this chapter the model calibration and the model validation procedure is presented. First a motivation to why model calibration and validation is needed is presented. Henceforth, the calibration procedure is described and which equations and parameters the minimization is based on. At last the implementation of the minimization in MATLAB and adequate functions are discussed.

Before the traffic model can be used for the implementation of VSL, the model needs to be calibrated with real-time data or simulated data from VISSIM. The objective of this procedure is to elaborate a macroscopic model of the infrastructure which describes the real traffic behavior. To get the best precision as possible the design parameters are specified within valid range (Papageorgiou et al., 2010).

The calibration optimizes the values of the following parameters:

- v_{fm}, the free speed on link m, that drivers assume if traffic is freely flowing
- α_m, a nonlinear relation between mean speed and density
- ρ_{cr}, the critical density per lane of link m, at which the traffic flow is maximal
- τ, relaxation time constant
- ν, anticipation factor
- κ, constant used to describe how the drivers adjusts their speed to what they anticipate will happen depending on what they see in the next segment
- δ, the term which describes the speed drop due to ramp-metering caused by on-ramp

To get the best possible fit, the boundary conditions in METANET should be the same as the simulation set up in VISSIM. After the calibration is finished, a validation of the parameter fitting is done by testing the new minimized parameters on the new data from simulation or real time data. If the minimized parameters also fit the new data, the model is realistic and usable. The calibration procedure can be seen in figure 5.1.

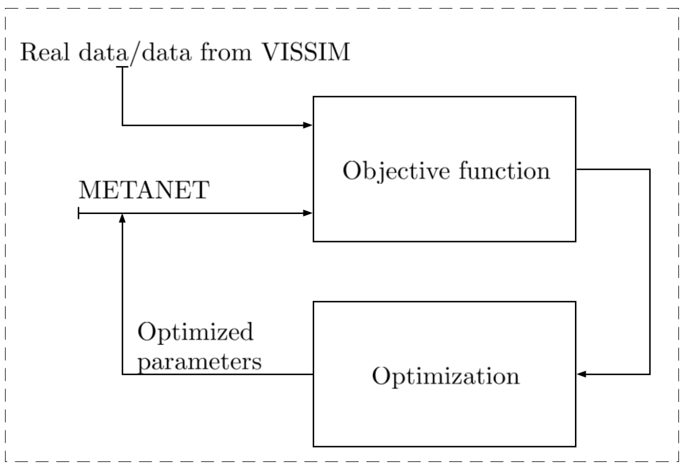


Figure 5.1: The calibration procedure

5.1 Parameter fitting

To estimate the parameters the least squares output error method is used, since the equations for the model are nonlinear both in parameters and variables. This will minimize the difference between the model and the real data (Cremer and Papageorgiou, 1981), as visualized in figure 5.2 below.

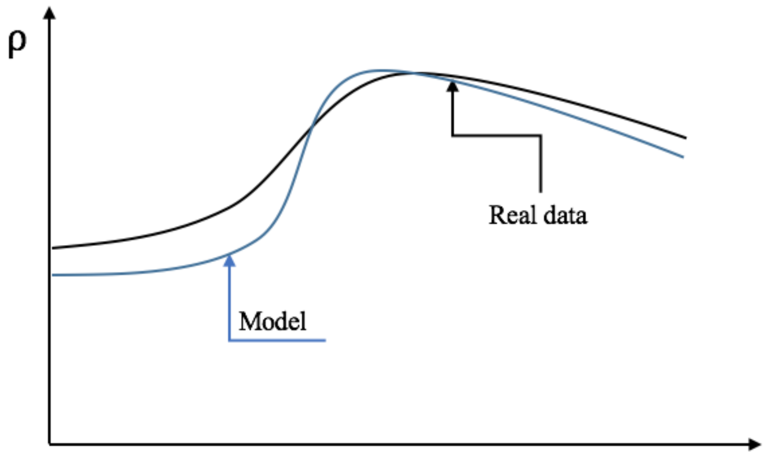


Figure 5.2: Example of the parameter fitting, the calibration is done by reducing the difference between the model and the real data.

The same equations as in METANET are implemented in a function script. The function takes the decision variables as input and returns the objective function to minimize. The objective function is the sum of the distance between the theoretical values from METANET and the simulated samples from VISSIM.

$$F_{obj} = \sum_{l=0}^{N_{samp}} \sum_{(m,i)\epsilon I_{all}} ((\hat{q}_{m,i}(l) - \tilde{q}_{m,i}(l))^2 + \xi(\hat{v}_{m,i}(l) - \tilde{v}_{m,i}(l))^2)) \tag{5.1}$$

where:

- N_{samp} is the amount of data samples
- I_{all} is the number of all links and segments
- $\tilde{q}_{m,i}(l)$ and $\tilde{v}_{m,i}(l)$ is the measured speed and flow
- ξ is a weight constant
- $\hat{q}_{m,i}(l)$ and $\hat{v}_{m,i}(l)$ is the simulated speed and flow

By choosing the model parameters appropriately, the objective function will be numerically minimized and a global minimum for the function will be found. Finding a global minimum can be difficult, because many of the methods finds a local minimum which does not necessarily have to be a global minimum. If a global minimum is found the model is fitted as good as possible.

5.2 Minimization method fmincon

The function fmincon in MATLAB is used to find minimum of constrained nonlinear multivarible function. To use the function fmincon initial values, x_0, for the parameters needs to be defined. Also, an upper boundary (ub) and a lower boundary (lb) needs to be defined. This means that the solution for the minimization of the parameters can be found in the range $lb \leq x \leq ub$ (where x is the optimized parameters). As mentioned before, it is possible to find a local minimum instead of a global minimum. If so, the initial values and/or the boundaries needs to be redefined.

The function fmincon also takes some constraints for static nonlinear equalities and inequalities as input. Since neither static nonlinear equalities or inequalities exists in this case those inputs are empty. The output from fmincon is a vector $\boldsymbol{x}$ with the values of the optimized parameters. The code for the minimization can be found in Appendix C and the code for the objective function can be found in Appendix D.

6 Controller design

In this chapter, the process of developing the controller design used is discussed. In order to design a controller with desired properties, a few steps needs to be followed. The first step is to linearize the METANET model, to be able to create a linearized state space model of the system. The state space model is then used to achieve an optimal control structure for the controller design. The controller is later used together with METANET and the COM interface to control the speed and density with VSL.

6.1 Linearization of METANET

Since METANET is a nonlinear model, a linearization must be done to be able to control the model. Linearization is a method where the system is assessed around a chosen point. The point is preferably a stationary point. The nonlinear system:

$$f_1 = \rho_{m,i}(k+1) = \rho_{m,i}(k) + \frac{T}{L_m \lambda_m}[q_{m,i-1}(k) - q_{m,i}(k)] \tag{6.1}$$

$$\begin{aligned} f_2 = v_{m,i}(k+1) = v_{m,i}(k) &+ \frac{T}{\tau}\{V[\rho_{m,i}(k)] - v_{m,i}(k)\} \\ &+ \frac{T}{L_m}[v_{m,i-1}(k) - v_{m,i}(k)]v_{m,i}(k) - \frac{\nu T}{\tau L_m}\frac{\rho_{m,i+1}(k) - \rho_{m,i}(k)}{\rho_{m,i}(k) + \kappa} \end{aligned} \tag{6.2}$$

$$f_3 = w_o(k+1) = w_o(k) + T(d_o(k) - q_o(k)) \tag{6.3}$$

The stationary point is defined by the point where $\rho_{m,i}(k+1) = \rho_{m,i}(k)$, $v_{m,i}(k+1) = v_{m,i}(k)$ and $w_o(k+1) = w_o(k)$. The stationary point is calculated with the fsolve command in MATLAB. To be able to calculate the stationary point some parameter must be set. In this case the density ($\rho_{m,i}$) is being set to the critical density (ρ_{crit}) as the goal is to keep the density close but lower than the critical density. The design parameters are taken from the calculations in the model calibration chapter. The stationary speed v_{stat} is then given by the MATLAB code. The code can be found in appendix E. The next step is to evaluate the Jacobian matrices A and B. The Jacobian matrix is all first order partial derivatives of the vector valued functions (6.1), (6.2) and (6.3). In matrix $\boldsymbol{A}$ the system is derived with respect of $\rho_{m,i}(k)$, $v_{m,i}(k)$ and $w_o(k)$. $\boldsymbol{B}$ is the system derived with the respect of the control variable u which gives the result:

$$\boldsymbol{A} = \begin{bmatrix} \frac{\partial f_1}{\partial \rho_{m,i}(k)} & \frac{\partial f_1}{\partial v_{m,i}(k)} & \frac{\partial f_1}{\partial w_o(k)} \\ \frac{\partial f_2}{\partial \rho_{m,i}(k)} & \frac{\partial f_2}{\partial v_{m,i}(k)} & \frac{\partial f_2}{\partial w_o(k)} \\ \frac{\partial f_3}{\partial \rho_{m,i}(k)} & \frac{\partial f_3}{\partial v_{m,i}(k)} & \frac{\partial f_3}{\partial w_o(k)} \end{bmatrix} \tag{6.4}$$

$$\boldsymbol{B} = \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \\ \frac{\partial f_3}{\partial u} \end{bmatrix} \tag{6.5}$$

The matrices are then evaluated at the stationary point $\rho_{m,i}(k) = \rho_{cr}$, $v_{m,i}(k) = v_{stat}$, $w_o(k) = 0$ and creates a discrete state space representation of the system. The discrete state space model is presented below where $\boldsymbol{x}$ is a vector consisting of $\rho_{m,i}$, $v_{m,i}$ and w_o and can be used later to create a controller. The output $\boldsymbol{y}$ is in this case the equilibrium speed V. $\boldsymbol{A}$ is the state matrix, $\boldsymbol{B}$ is the input matrix, $\boldsymbol{C}$ is the output matrix and $\boldsymbol{D}$ is the feedthrough matrix.

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}u(k) \tag{6.6}$$

$$\boldsymbol{y}(k) = \boldsymbol{C}(k)\boldsymbol{x} + \boldsymbol{D}(k)u(k) \tag{6.7}$$

6.2 Controller Implementation

There are different types of controllers that can be used to regulate a system. Which controller that suits a specific system depends on the systems demands. The use of controllers can be in our daily life, e.g. to regulate hot and cold water when taking a shower, cruise control in car or to control variable speed limits.

The linear-quadratic regulator (LQR) takes advantage of a state-space approach to analyze systems. By using full state feedback, it is possible to stabilize the system (Nasir et al., 2008). The controller is seen as one of the optimal control techniques because it makes optimal control decisions, based on the control input and the states of the dynamical system (Prasad et al., 2012). LQR is a relatively simple controller, compared to other controllers, and is therefore easy to implement.

MATLAB has a built in function, lqr, that is used to be able to create a LQR design. For a discrete time state-space system, the function dlqr (discrete lqr controller) is used to create a linear-quadratic state-feedback regulator design. When calling the dlqr, inputs are given by the A, B and Q_e matrices and the R_e parameter. The A and B matrices are computed from the linearization. The eigenvalues of Q_e matrix are ranking the variables importance. A small eigenvalue near zero means that the parameter is of low interest to optimize. In this case, the density is of most importance meanwhile the speed and queue length is of lesser importance. The R_e parameter is a attenuation parameter to make the changes in limits in VSL more smooth. Even though R_e smooths out the VSL there can still be noise left in the VSL model. The dlqr function gives a vector k_{lqr} as an output. To create the control variable and speed limit the following equation is implemented:

$$u(i,k) = -k_{lqr}[\rho_{m,i}(i,k) - \rho_{cr}; v_{m,i}(i,k) - v_{stat}; w_o(k) - 0] + V_{limit} \tag{6.8}$$

Where $\rho_{m,i}$, $v_{m,i}$ and w_o are given by METANET or live measurements. The critical density, ρ_{cr}, is calculated according to chapter 5 and the v_{stat} is calculated in the linearization.

6.3 Filtering

The LQR controller gives a control variable with much unwanted noise. To get a more clear signal with not so fluctuated behavior the controller can be combined with a filter.

$$G(s) = \frac{\frac{1}{\tau_f}}{s + \frac{1}{\tau_f}} \tag{6.9}$$

Equation (6.9) describes a low pass filter that can be used to filter signals and get rid of the fast changing behavior. A typically good value for the time constant τ_f is $\tau_f = 0.1$. The filter is converted to a discrete state space model with the state matrix a_f, the input matrix b_f, the output matrix c_f and the feedthrough matrix d_f. These matrices is used in the following equations to be applied on the control variable u to filter it and create the output u_f.

$$x_f(i,k+1) = a_f x_f(i,k) + b_f u(i,k) \tag{6.10}$$

$$u_f(i,k) = c_f x_f(i,k) + d_f u(i,k) \tag{6.11}$$

6.4 Controller in METANET

In METANET the controller was implemented in MATLAB as showed in the steps above. The equations for the controller was combined with equations (2.2)-(2.10) from chapter 2 to create the model. To make the speed limits more realistic the control variable is rounded down to commonly used speed limits according to the values in table 6.1.

Table 6.1: Rounding of speed limits in METANET

Control variable	Speed limit [km/h]
$u_f \geq v_{fm}$	120
$v_{fm} > u_f \geq 90$	90
$90 > u_f \geq 70$	70
$70 > u_f \geq 50$	50
$50 > u_f$	30

In METANET the controller is only active if the density $\rho_{m,i}$ is higher than the critical density ρ_{cr}. This means that the speed limit is only changed with the control algorithm if the density is high enough.

6.5 Controller in COM interface

Several data collection points were placed out along the road with a distance of 500 m in VISSIM. When implementing the controller in VISSIM the variables $v_{m,i}$ and $\rho_{m,i}$ are collected dynamically in the simulation as mentioned in section 4.4. The METANET model is linearized according to section 6.1 with the optimized parameters according to the parameter fitting as mentioned in chapter 5. The LQR controller is implemented as described in section 6.2. From the COM interface simulations are run step by step so the speed limits can be controlled dynamically. If the density is higher than the critical density the speed limits are changed to the speed which is calculated by the controller, in other words the speed is not controlled if the density is below the critical density. Before the speed limits are changed, the calculated speed is rounded down to a commonly used speed limit which makes the speed limits more realistic. The rounding in the COM interface is executed according to table 6.2. The implemented control logic works in the following manner, the segments are analyzed in decreasing order meaning the last segment is analyzed first and thereafter the penultimate segment back to the first. If the speed limit in a segment is reduced all the downstream segments are reduced as well to delay the congestion from occurring.

Table 6.2: Rounding of speed limits in COM interface

Control variable	Speed limit [km/h]
$u \geq v_{fm}$	90
$90 > u \geq 80$	80
$80 > u \geq 70$	70
$70 > u \geq 60$	60
$60 > u \geq 50$	50
$50 > u \geq 40$	40
$40 > u$	30

7 Implementation and results

In this chapter, the selected case study on the highway E6/E20 Mölndal is presented in section 7.1. After that, results from the implementation of the METANET model is introduced, both with boundaries from VISSIM and with selected boundary conditions. In the last section, results from two different simulation scenarios in VISSIM, conducted through the COM interface from MATLAB, is presented.

7.1 Mölndal E6/E20

For the case study, the area Mölndal E6/E20 was chosen. The area is shown in figure 7.1.1[4] where an overview of Gothenburg is shown to the left and the selected road is marked in red to the right. This area has several VSL-signs, but is still a conflicted road in the Gothenburg area. Information from the Swedish Transport Administration in Gothenburg showed that this section has a high level of congestion during rush hours, which is the reason why this area is investigated. During the study visit at the Swedish Transport Administration, the screens in the control room displayed that E6/E20 Mölndal were one of the areas with highest traffic activity and congestion.

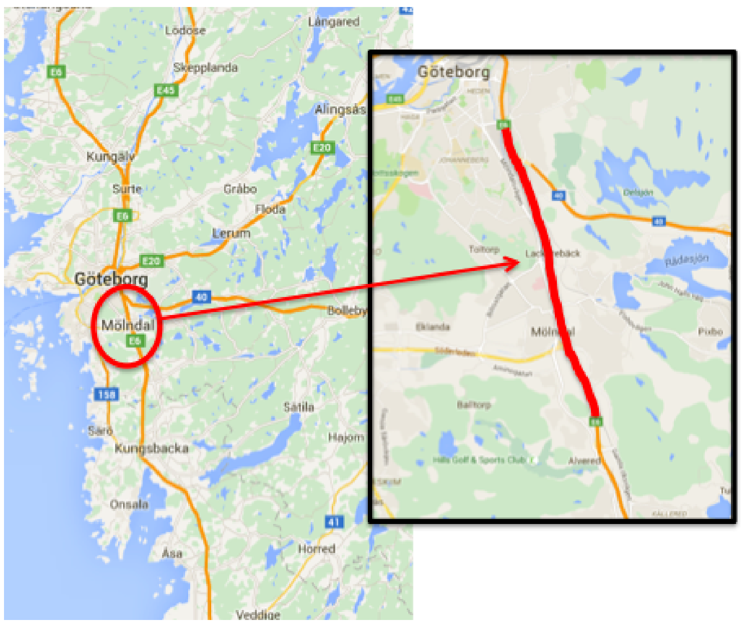


Figure 7.1.1: A map of the selected area and road Mölndal E6/E20

A simplified model of the selected distance E6/E20 in the Mölndal area was built up in VISSIM to represent the real traffic network. A part of this model is displayed in figure 7.1.2. Links were constructed resembling the real links in Mölndal. A real map over the area around Gothenburg is loaded in to VISSIM as background and the links are built along the E6/E20. In the simplified case only one on-ramp is connected to the link in difference from the real link. The simulated route stretches about 3-7 km, from the Liseberg area to Torrekullamotet with Mölndal situated in between, depending on the different simulation scenarios in sections 7.2-7.4. The different scenarios were simulated with different conditions, such as constant speed limits and constant traffic flow, upstream congestion etc, and results were saved and analyzed both for the uncontrolled scenario and the scenario with VSL control. In all scenarios the road was divided into 500 m long segments which were evaluated and analyzed.

[4]Maps collected from (Google maps, 2016)

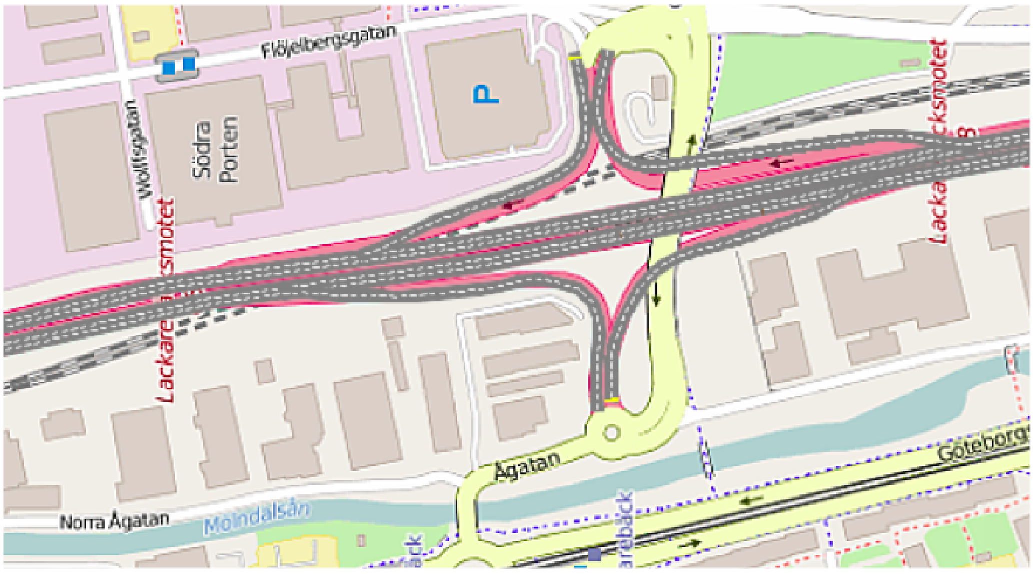


Figure 7.1.2: An overview of the modeled road in Mölndal

7.1.1 Collection of real time data

Real time and historical data was collected from the website (Vägverket, 2016). The purpose was to analyze during which time period of the day the traffic activity were most dense, and note the values of the parameters during that interval. The available data is possible to collect from many different roads in or in connection to Gothenburg. Flow, speed, occupancy, confidence and headway is the available parameters and it is also possible to show data from specific lanes and for different types of vehicles. Another functionality is to choose different time intervals and different time resolutions.

The chosen section for the real time data collection was "E6 Åbro mot Kållered", since Mölndal E6/E20 is a part of this road. To get an overview of which day of the week that had the highest traffic activity, the mean flow and the mean speed was observed during several weeks. The result showed that the day with the highest mean flow and mean speed was varying from week to week but it was in general higher during week days. Because of that, the day for the real time data collection was chosen to be a week day and was randomly chosen to April 20th 2016. The day was divided into three hour intervals to be able to analyze which time period that had the highest traffic activity. The results from the data collection is shown in table 7.1.

Table 7.1: Real time data collection from "E6 Åbro mot Kållered"

Collected data			
Date	Time period	Mean flow $[veh/h]$	Mean speed $[km/h]$
20/4	00:00-23:59	1527	92.3
20/4	00:00-03:00	66	90.6
20/4	03:00-06:00	154	93.0
20/4	06:00-09:00	1672	95.8
20/4	09:00-12:00	1641	92.6
20/4	12:00-15:00	2076	92.5
20/4	15:00-18:00	4349	88.5
20/4	18:00-21:00	1577	94.2
20/4	21:00-23:59	565	91.5

From table 7.1 it is clear that the time period with the highest traffic activity is between 15:00 and 18:00. This time period had almost three times as high mean flow, compared to the average flow of the whole day. It also had the lowest mean speed and was the only time period that had a mean speed lower than 90 km/h.

7.2 METANET - boundaries from VISSIM

In this section the design parameters in METANET is optimized to get a model that fits the simulated data from VISSIM. After that, the controller presented in chapter 6 are implemented on the calibrated and validated model. Two simulations in VISSIM are done, where the first simulation is used to fit the model parameters and the second simulation is used to validate the model.

When the selected link was built up in VISSIM, the next step was to develop a simulation with congestion. Input such as flow, demand and speed limits were used to create high activity in the network. The chosen data for the simulations is presented in table 7.2. The aim was to get data from VISSIM that could be used to find the design parameters to fit the METANET model to simulated data from VISSIM. To get a good fit for the model, it is necessary to have data from a simulation with an occurring congestion.

Table 7.2: Selected input values to create congestion in VISSIM.

Simulation setup in VISSIM				
Simulation 1	Time [sec]	0-900	900-1200	1200-2000
	Flow [veh/h]	4000	8000	3000
	Demand on-ramp [veh/h]	1000	1000	1000
	Speed limit last segment [km/h]	120	50	120
Simulation 2	Time [sec]	0-900	900-1200	1200-2000
	Flow [veh/h]	4000	8000	3000
	Demand on-ramp [veh/h]	1400	1400	1400
	Speed limit last segment [km/h]	120	50	120

A high speed for the vehicles during the simulations was desirable, since the focus in the case study is on highways. A speed limit was set to 120 km/h in the beginning of the first segment in each of the two simulations, due to the aim in creating a simulation resembling a case with no set speed limit as in the METANET model. When no speed limit was set in VISSIM, the vehicles drove in a low speed (40-50 km/h) hence a high speed limit was necessary to create a case representing the scenario where drivers drive as fast as they desire. The length of the link in the simulation runs were approximately 3 km. The link was divided into six, 500 m, segments and the results were collected and saved from each segment. In the end there was a short segment, excluded from the results, only existing to lower the speed limit in the end of the link. The purpose with this segment was to reduce the speed to create a congestion. On the fifth segment an on-ramp, with a flow input entering the main link, was connected to the link to make a more realistic model of a road. An attempt to design a model that fits a simulation with more segments and longer distance were also done, but this gave inadequate results for the segments in the middle of the link.

To verify that congestion occurred during the simulations, fundamental diagrams was plotted. Figure 7.2.1 below represent the fundamental diagram for the first simulation run. Simulation run 2 showed a higher level of congestion in the fundamental diagram compared to the fundamental diagram of simulation run 1. This indicates that simulation run 2 reached a higher density level but other than this the diagrams were quite similar.

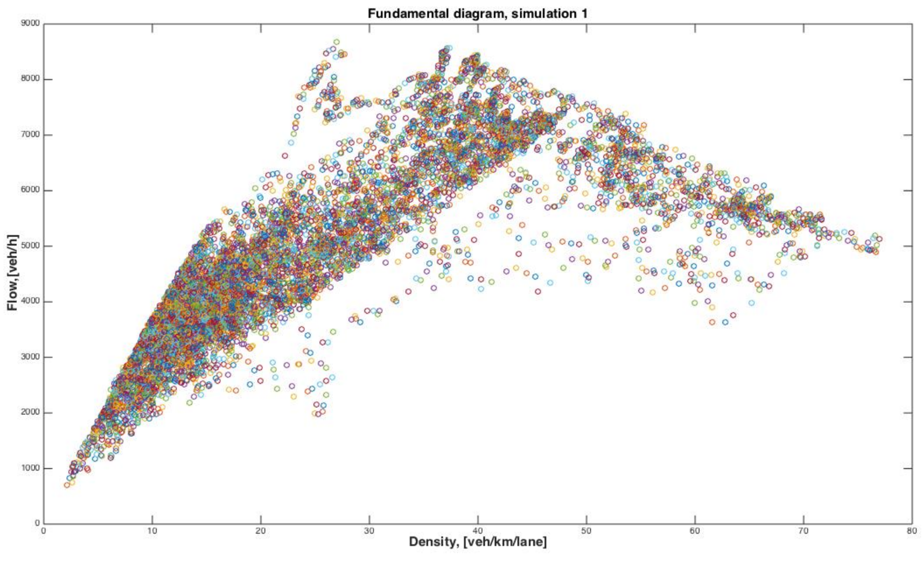


Figure 7.2.1: Fundametal diagram for simulation 1

7.2.1 Calibration

After collecting data from VISSIM the following step was made to identify optimal values for the design parameters, making the model fit the data as good as possible, by using the minimization procedure presented in chapter 5. The first and the last of the six segments were used as boundary conditions and the results from VISSIM were used as values. In table 7.3 the optimal values for the parameters found for simulation run 1 is presented.

Table 7.3: Design parameters for simulation 1

Parameters							
	v_{fm}	α_m	ρ_{cr}	τ	ν	κ	δ
Simulation 1	125.9805	1.2336	35.0087	0.0020	47.4938	50.0000	0.0300

To make sure that the calibration of the model was successful, i.e. that the model fits the data good enough, a comparison for each segment was done. The figures below shows the level of resemblance in the comparison between simulation run 1 and the produced model. In the comparison density and speed is taken into consideration. Figure 7.2.2 plots the density for simulation run 1 and for the produced optimized model. It can be observed in figure 7.2.2 that the model fits the simulation better in the lower density areas meanwhile not reaching the same level as the simulated data in the high density region. The same results can be observed for the mean speed in figure 7.2.3 where the mean speed in the model does not reach as low levels as the simulated mean speed. In VISSIM the variation in mean speed is more visible compared to the more stable mean speed in the model. In VISSIM some noise was included in the simulation which explains the fluctuating behavior in mean speed. The plots for rest of the segments displayed approximately the same results, with the exception of the first and the last segment which were set as boundary conditions, hence did not change. To summarize, the figures indicates a good fitting and the optimized values for the design parameters should be validated with a new simulation.

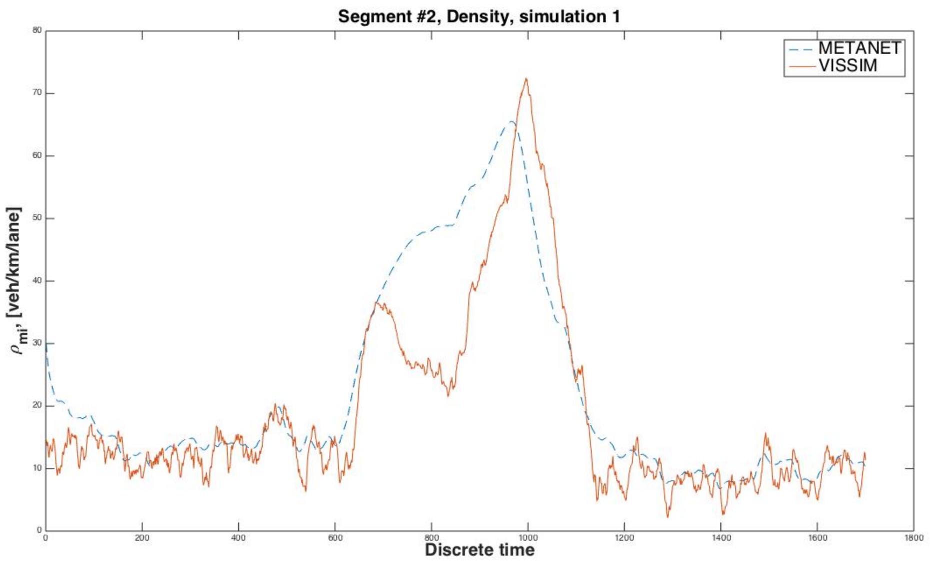


Figure 7.2.2: Density in segment 2, simulation run 1 in VISSIM and best calibrated model in METANET

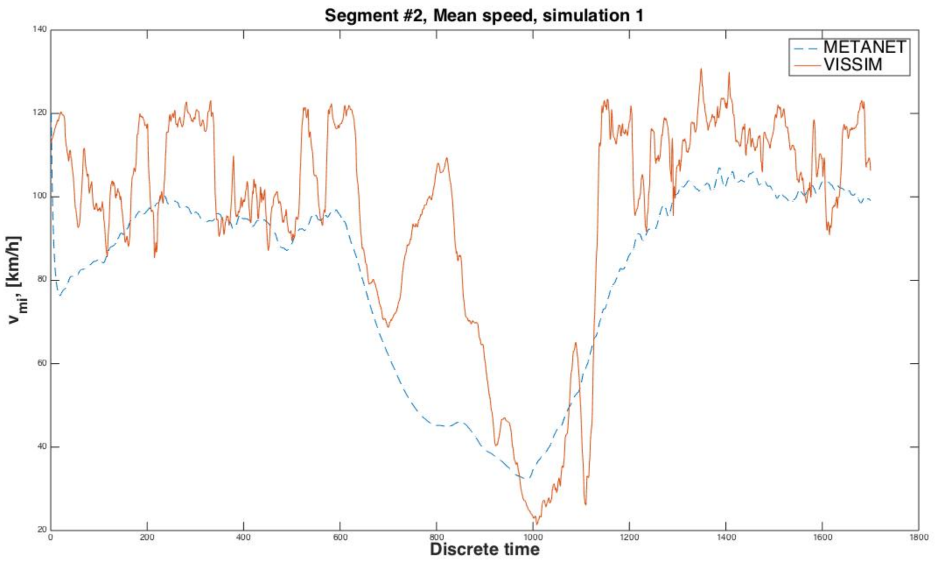


Figure 7.2.3: Speed in segment 2, simulation run 1 in VISSIM and best calibrated model in METANET

7.2.2 Validation

To validate if the model is good the estimated design parameters are tested on simulation run 2 with different initial conditions. The model and optimized design parameters is good if the results from VISSIM and METANET do not differ much i.e the curves are quite similar in the validation case as well.

Figure 7.2.4 below shows the density in the same segment as in the calibration. In this case, there is a bigger difference between the model and simulated data during the highest density than in the previous simulation. An explanation can be that the design parameters optimized for the first simulation run, with a lower level of congestion, does not make the density for the model reach as high as desired. Figure 7.2.5, for speed, shows that the mean speed in the segment stays low longer in VISSIM than for the model during the congestion phase. Still, the curves follow each other relatively good. This made it possible to implement VSL on METANET and estimate what effects this can have on the link.

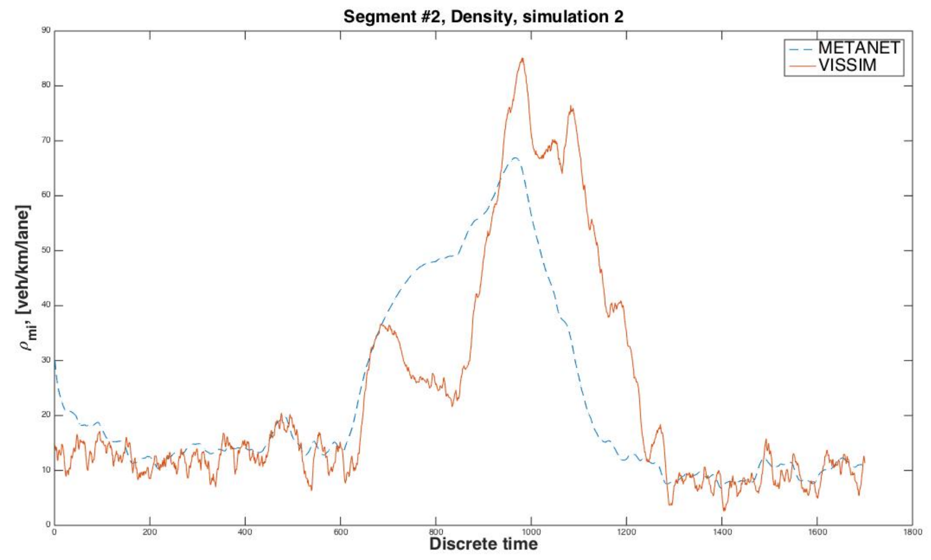


Figure 7.2.4: Density in segment 2, simulation run 2 in VISSIM and the validated model in METANET

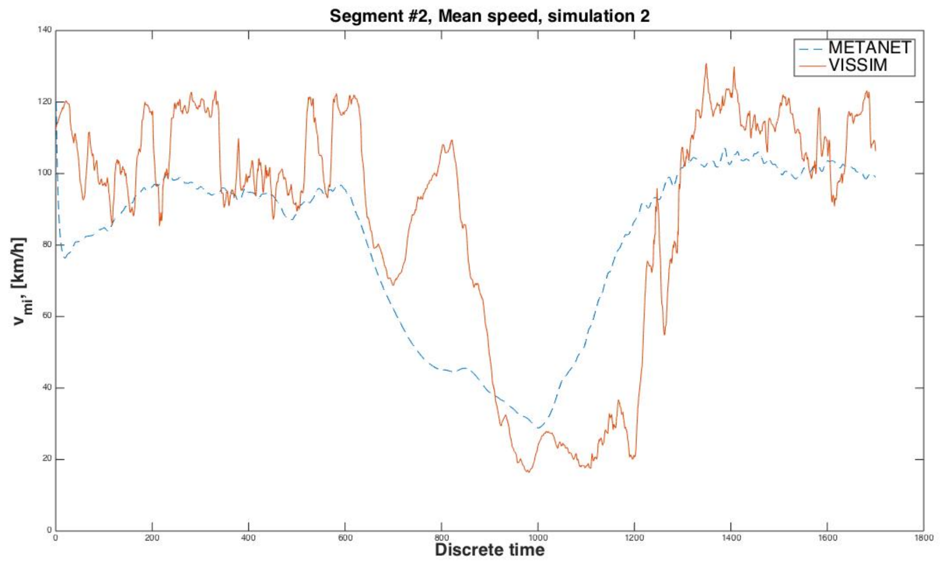


Figure 7.2.5: Speed in segment 2, simulation run 2 in VISSIM and the validated model in METANET

7.2.3 Scenario 1

Following the validation the next step was to implement the controller suggested in section 6.2. The first scenario to test the controller is on the model produced with simulation run 1. The boundary conditions were chosen as the data from the simulation.

The purpose with the control variable is to decrease the speed limit whenever the density exceeds the critical density. During the simulation, the initial speed limit was set to 120 km/h. In figure 7.2.6, the control variable is represented by the yellow line, which is constantly set to 90 km/h. Although, the speed limit is not set to 90 km/h all the time. In the same figure as the speed limit, $V[\rho]$ for both the uncontrolled and controlled case is shown. If the controlled case has a density lower than the critical density, $V_{uncontrolled} = V_{controlled}$. The blue curve in the figure fluctuates a lot, which can be due to that the density is around the critical density, which activates and deactivates the controller during short time periods. In real life, this would be solved by not letting the controller switch on/off that fast which would result in a more consistent curve.

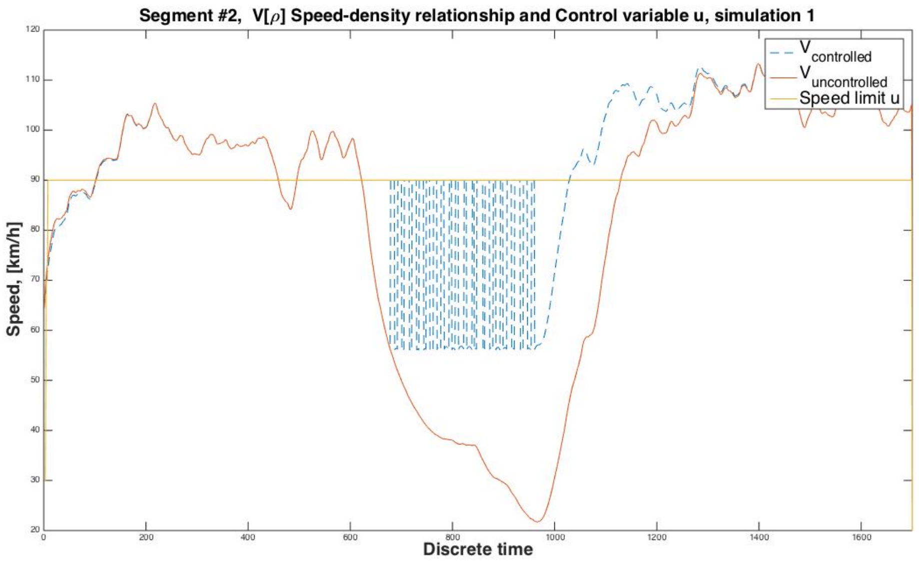


Figure 7.2.6: The control variable, $V_{controlled}$ and $V_{uncontrolled}$

The figure 7.2.7 shows how the density is affected when the controller is used. The optimized value for ρ_{cr} was around 35 $veh/km/lane$. When the density is getting close to the value for the critical density, the blue curve stays around that value. As the picture displays, the curve for VSL lowers the density when it exceeds the critical density (ρ_{cr}) during the same time as the control variable is activated shown in the previous picture.

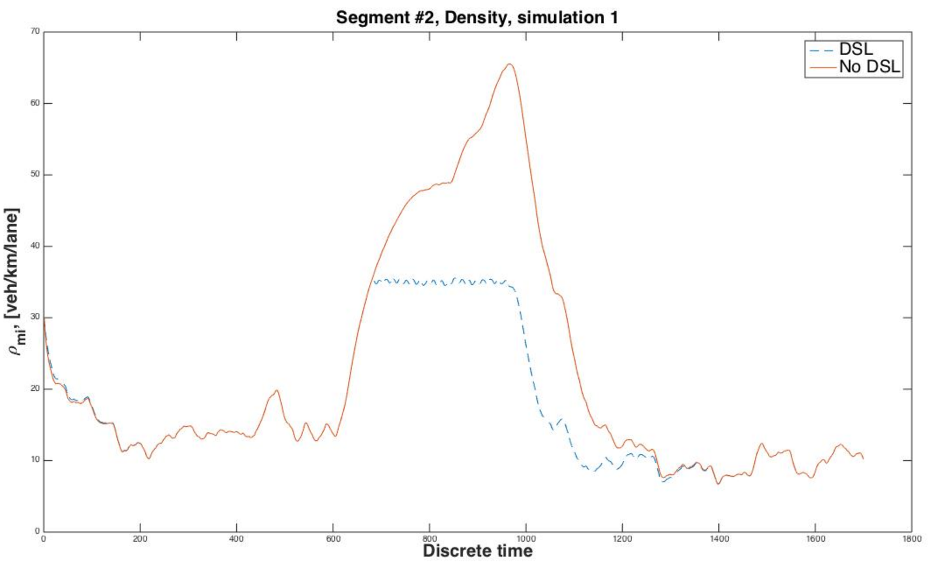


Figure 7.2.7: Density in segment 2, when implementing VSL

Additionally, to show that the model is similar to the simulated data, and how VSL can affect the model, 3D plots were made displaying the density. Figure 7.2.8 below shows the simulated data from VISSIM. The red area is indicating a high level of congestion. In figure 7.2.9, the uncontrolled model that emulates the simulation is shown. In this figure, it seems like the density is lower during the congestion phase than in the figure with simulated data since the red area is smaller. The simulated data also have more noise added which makes that plot fluctuate more than the plot for the model. Aside from those two differences, the two plots look similar. In figure 7.2.10 the controller is implemented. It is clear that VSL has made improvements in the whole link during the period with congestion.

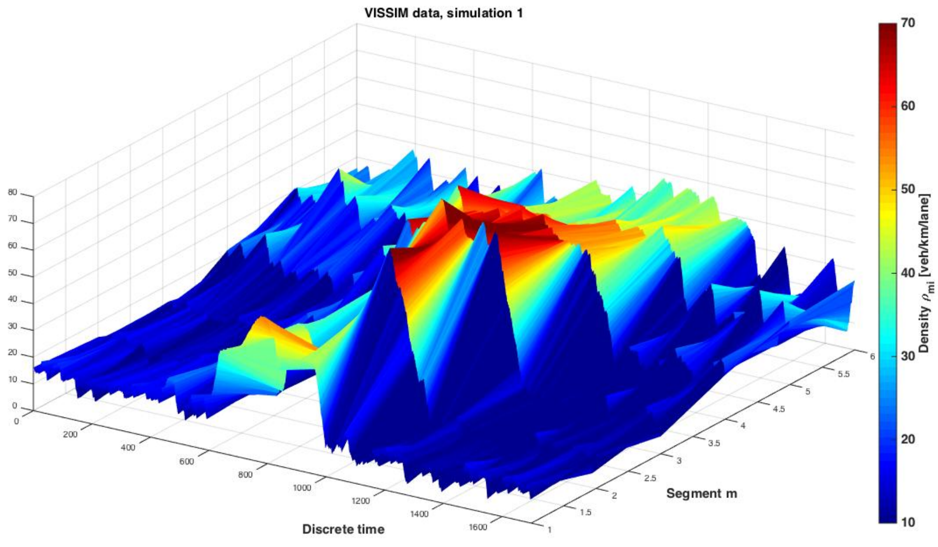


Figure 7.2.8: 3D density plot, simulation run 1, data from VISSIM

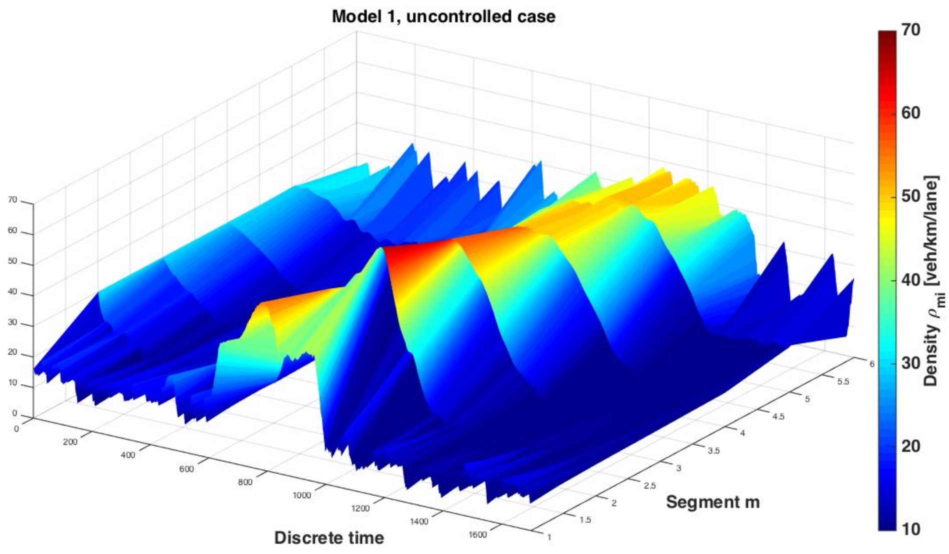


Figure 7.2.9: 3D density plot, uncontrolled case, calibrated model.

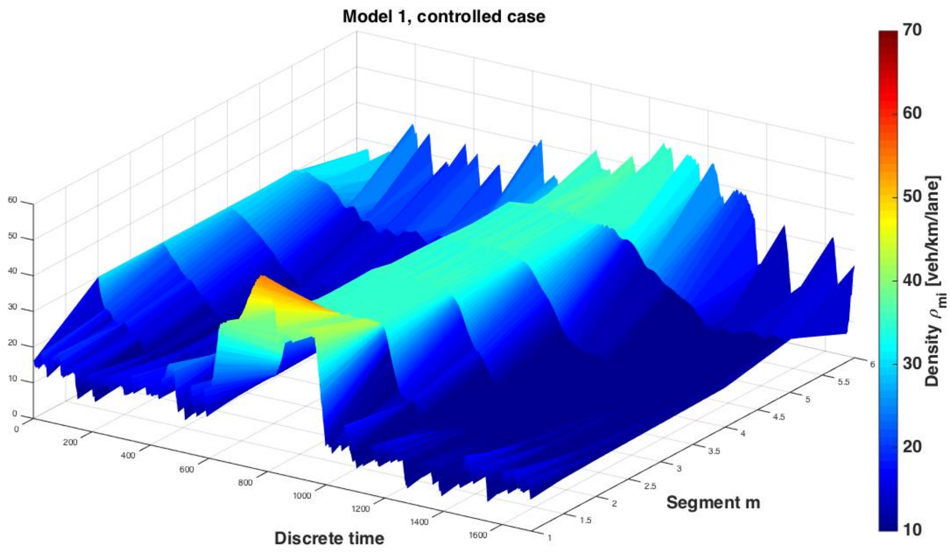


Figure 7.2.10: 3D density plot, controlled case, calibrated model.

7.3 METANET - selected boundary conditions

In this section selected boundary conditions is used to generate congestion, both propagating backwards and propagating forward for the controlled as well as uncontrolled cases, similar to the cases in section 2.3. The number of segments is chosen to 14 which makes the link 7 *km* long. The time horizon is chosen to 1000 *time units*.

7.3.1 Scenario 2

In scenario 2, a congestion that propagates forward were modeled. The boundary conditions were chosen according to table 7.4.

Table 7.4: Boundary conditions for a congestion propagating forward

Scenario 2			
Time interval	0-450	450-700	700-1000
Incoming flow $q_{m,1}(k)$	3000	4500	3000
Incoming speed $v_{m,1}(k)$	110	110	110
Outgoing density $\rho_{m,14}(k)$	25	25	25

Table 7.5: Selected design parameters

v_{fm}	α_m	ρ_{cr}	τ	ν	κ	δ
125	1.1867	31.1541	0.006	62.0642	20	0.0400

In reality, the traffic flow is not constant, which makes the exact values on the boundary conditions unrealistic. To create a more realistic scenario, random numbers were added to the boundary condition value. This was implemented according to the following line in MATLAB:

$$+10 * \text{rand} - 10 * \text{rand}$$

In Table 7.1, the mean flow for the time period between 15:00-18:00 is 4349 veh/h. Since this is a measure for the mean flow, the real flow can be higher during a part of this period. The value chosen for the flow to create a congestion was 4500 veh/h, which is similar to the flow at E6/E20 Mölndal during rush hours.

Figure 7.3.1 shows a congestion propagating forwards during the period when the incoming flow is higher. The critical density is chosen to ~31.15 $veh/km/lane$. In the figure 7.3.1 the density exceeds 35 $veh/km/lane$ and if VSL was used the controller would activate and decrease the speed limit in the red areas in the figure. For the uncontrolled case the red area in the figures is larger than for the controlled case (figure 7.3.2), where a red area is almost not seen. In the controlled case, the density stays below the critical density which makes the peaks not as high as in the uncontrolled case.

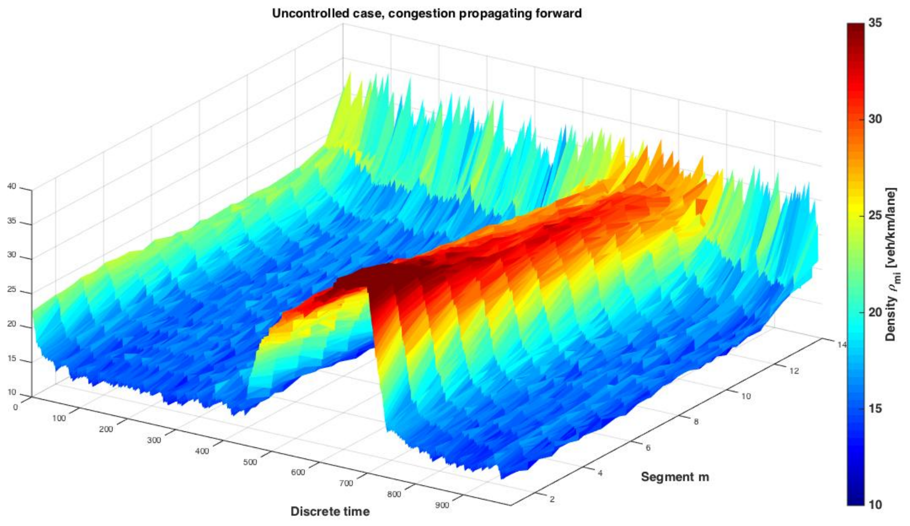


Figure 7.3.1: Density for the uncontrolled simulation when the congestion propagates forward, from a 3D view

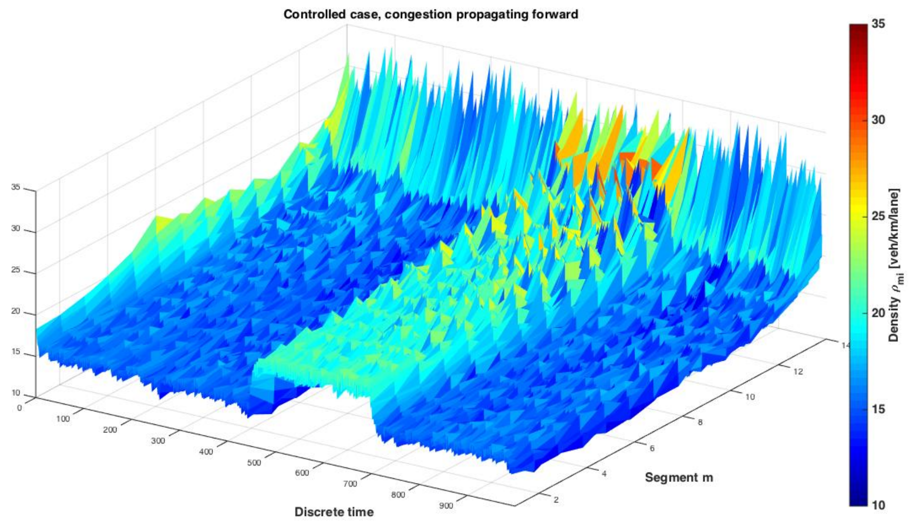


Figure 7.3.2: Density for the controlled simulation when the congestion propagates forward, from a 3D view

The density is equal for both the controlled and uncontrolled case in the figure 7.3.3. When the density increases and the curves deviates from each other, the controller activates and the speed limit is set to 90 km/h which is shown in figure 7.3.4. The controller is only active during the time that the red curve deviates from the blue curve in figure 7.3.3. This means that the speed limit is not set to 90 km/h when the curves overlaps, during this time it is free speed.

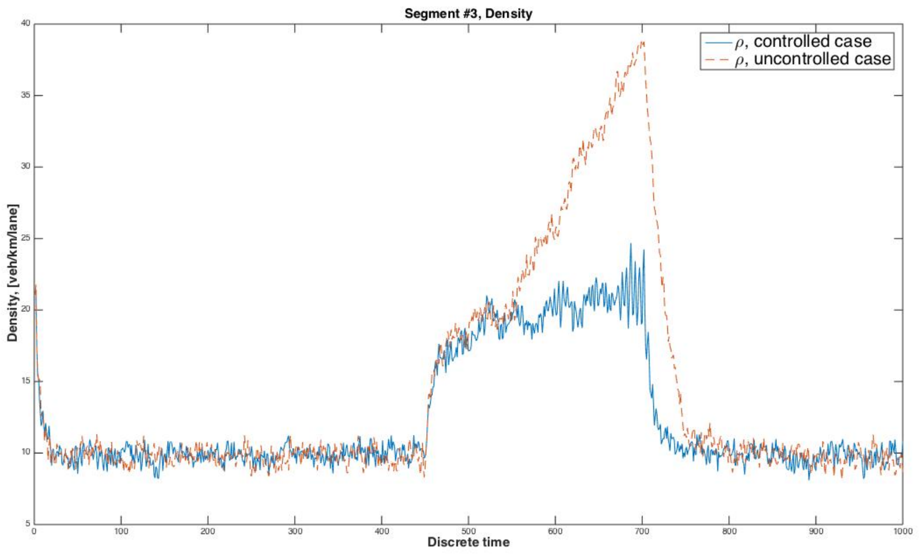


Figure 7.3.3: Density in segment number 3 for the congestion that propagates forward

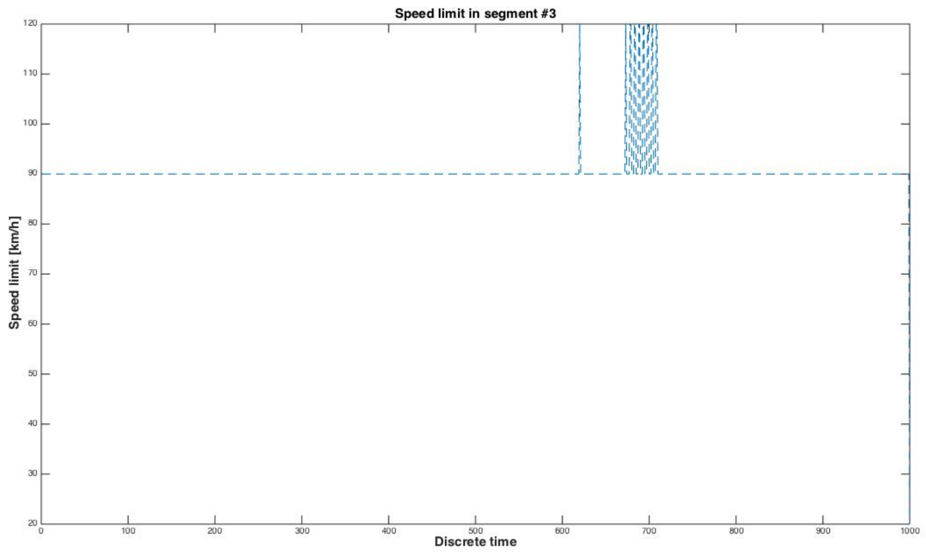


Figure 7.3.4: Speed limit in segment number 3, congestion propagating forward

7.3.2 Scenario 3

In scenario 3, a congestion propagating backwards were modeled. In the table 7.6 the boundary conditions used are presented.

Table 7.6: Boundary conditions for a congestion propagating backwards

Scenario 3			
Time interval	0-450	450-700	700-1000
Incoming flow $q_{m,1}(k)$	3000	3000	3000
Incoming speed $v_{m,1}(k)$	110	110	110
Outgoing density $\rho_{m,14}(k)$	ρ_{cr}-5	ρ_{cr}+15	ρ_{cr}-5

Table 7.7: Selected design parameters

v_{fm}	α_m	ρ_{cr}	τ	ν	κ	δ
125	1.1867	31.1541	0.006	62.0642	20	0.0400

Figure 7.3.5 below represents the uncontrolled case. It can be seen that the highest peak is more dense and have a higher density level overall, compared to the controlled case (figure 7.3.6). The controlled case has a lower density level in general, which can be seen when comparing the red areas in the figures. For the uncontrolled case the congestion is more extended than for the controlled case where the density level goes low quicker. As mentioned earlier, for the case where the congestion is propagating forward, the controller is also here only activated during the time that the density exceeds the critical value.

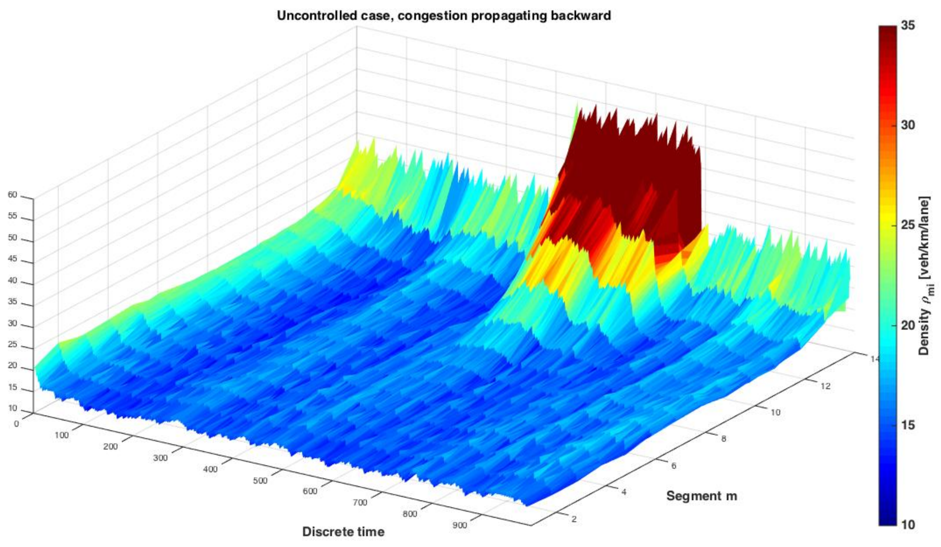


Figure 7.3.5: Density for the uncontrolled simulation when the congestion propagates backwards, from a 3D view

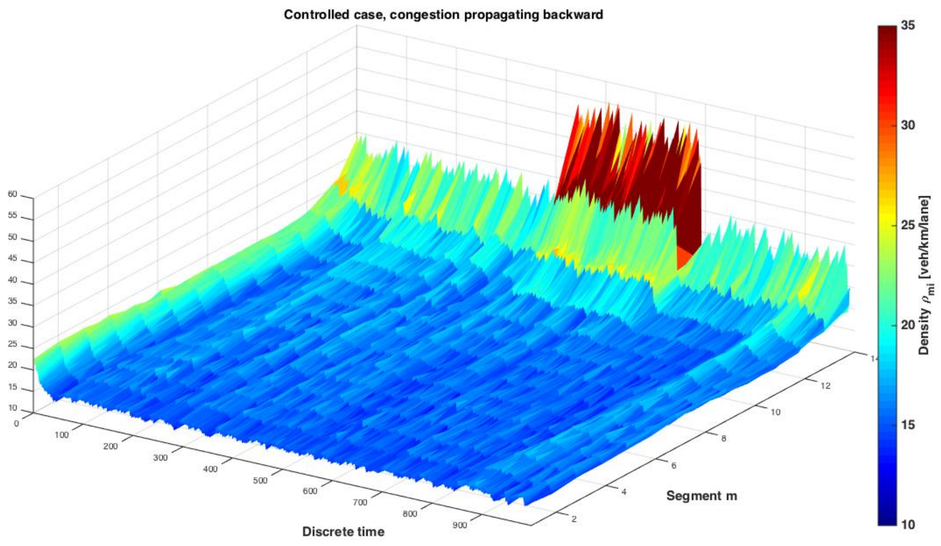


Figure 7.3.6: Density for the controlled simulation when the congestion propagates backwards, from a 3D view

7.4 COM interface - VISSIM

In the following section results from simulations in VISSIM conducted from the COM interface are presented. Two scenarios were designed with the input values that can be found in table 7.8. In each scenario, first a simulation run were conducted without any control measures implemented. Secondly, another simulation run were conducted with the same input but with the controller selected in chapter 6 implemented. The plots presented in this section comes from a link that consists of eight segments. The first seven segments are all 500 m long and the eighth segment is shorter. This segment is implemented in order to reduce the speed drastically to model a congestion and is not included in the results. An on-ramp is connected to segment 5 with a flow entering the segment. The critical density for the link is 31 $veh/km/lane$. Due to the fact that the network is completely empty when the simulation starts, the result starts recording after 300 *simulation seconds*. To clarify, the interval 0-600 corresponds to 300-900 in the simulation, and so on.

Table 7.8: Selected input values to create scenarios for the VSL control design

Simulation cases, VISSIM, COM interface				
Scenario 5	Time [*simulation sec*]	0-600	600-900	900-1700
	Flow [veh/h]	9000	9000	4500
	Demand on-ramp [veh/h]	1000	600	600
	Original speed limit [km/h]	90	90	90
	Speed limit in last segment [km/h]	90	90	90

7.4.1 Scenario 4

In scenario 4 simulation input values from table 7.8 were implemented. These setups simulates a congestion propagating backwards in the link with a low speed limit of 40 km/h in the last segment. The inflow of cars to the link and from the on-ramp is constant. In figure 7.4.1 and figure 7.4.2 it can be observed that a congestion with a density of 60-75 $veh/km/lane$ is created in segments 1-5 at the time 800-1700 with the probable cause of to high inflow from the on-ramp connected to segment 5. The density also exceeds critical density in segment 5-8 from time 400, which can be observed in figure 7.4.2. The cause of this small increase in density is the lowered speed limit in the last segment.

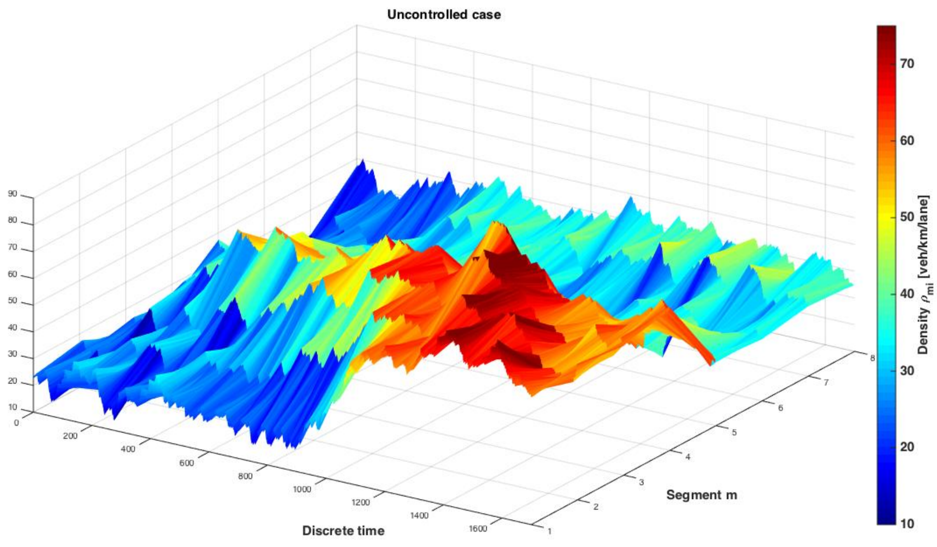


Figure 7.4.1: 3D density plot, scenario 4, uncontrolled.

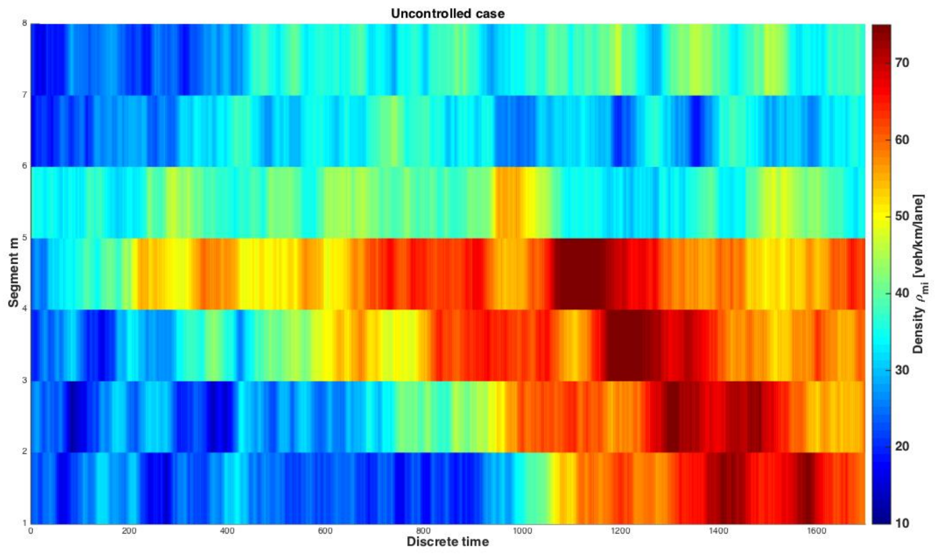


Figure 7.4.2: 2D view density plot, scenario 4, uncontrolled.

In figure 7.4.3 and 7.4.4 the controller has been implemented. During the highest level of congestion, i.e. the red areas, the density has been decreased in comparison to the uncontrolled case since the red color is not as intense. It can also be seen that the density in the last segments are somewhat the same or even higher than in figure 7.4.2. The aim with the controller is to accomplish a more homogeneous traffic flow. Since the variations in density for the controlled case is smaller, the controller has improved the traffic flow.

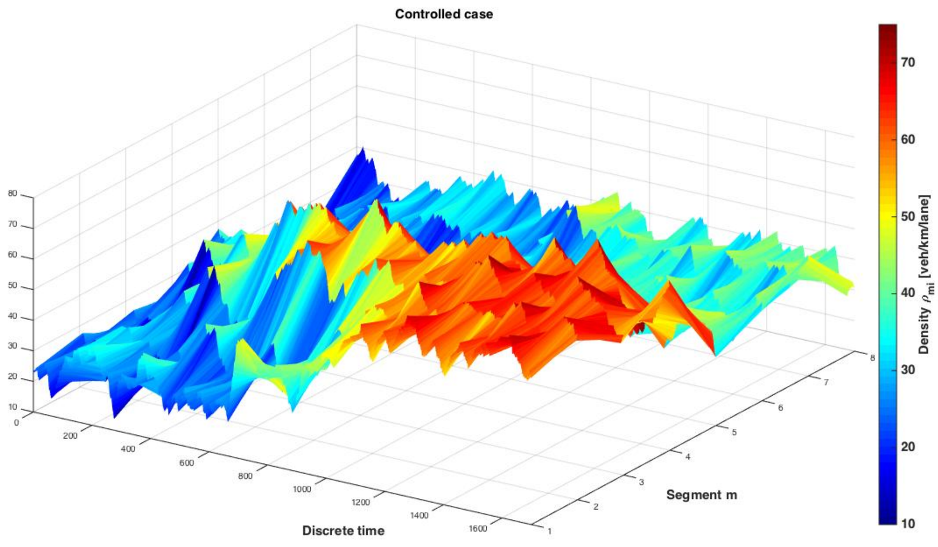


Figure 7.4.3: 3D density plot, scenario 4, controlled.

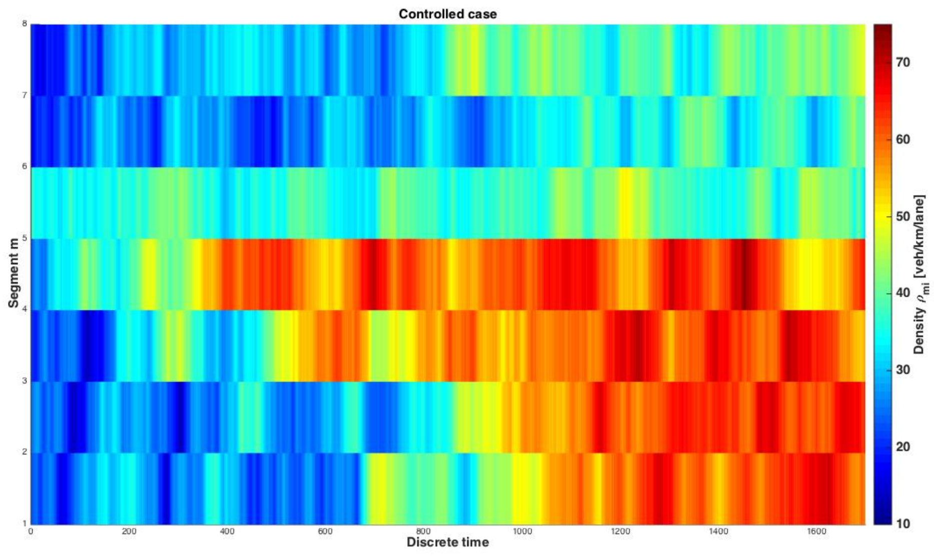


Figure 7.4.4: 2D view density plot, scenario 4, controlled.

Figure 7.4.5 shows how the density in segment 5 varies over time. The blue curve represents the uncontrolled case and the red curve represents the controlled case. The red curve touches or is below the blue curve for almost every time except for two minor periods, about 1100-1300 and 1600-1700. The controller is keeping the density around the critical value which delays the congestion. During the congestion phase for the uncontrolled case, the density reaches to ∼55 *veh/km/lane* while the controlled case only reaches ∼50 *veh/km/lane*. The difference between the highest and lowest density value is more visible for the blue curve, in other words the density is more stable when the VSL controller is implemented.

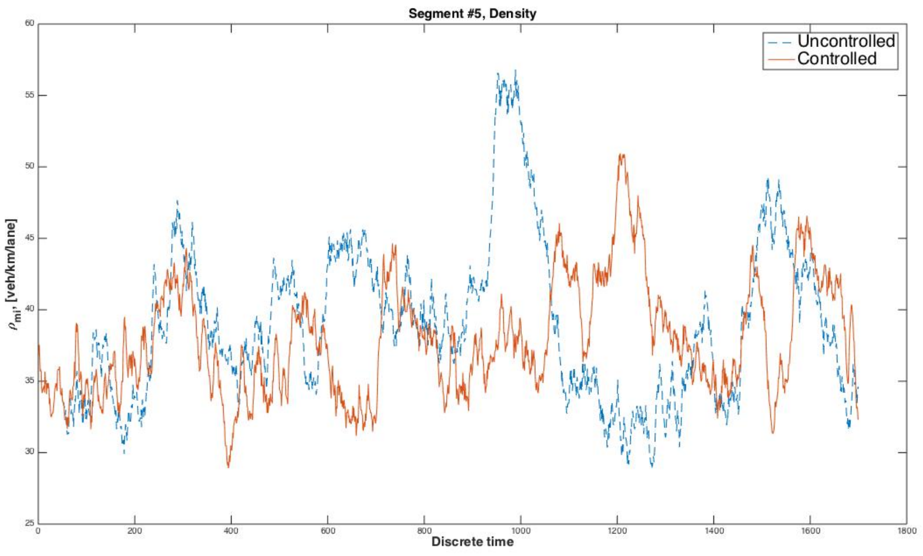


Figure 7.4.5: Density plot, scenario 4, segment 5.

Without VSL, the mean speed is around the speed limits at first which is indicated by the red areas in figure 7.4.6 and 7.4.7. During the same period that the density peaks, the mean speed drops down to about 20-30 km/h, described by the blue area. In the segments ahead of segment 5, where the on-ramp is located, there are some red and yellow areas implying that the mean speed is around the speed limits.

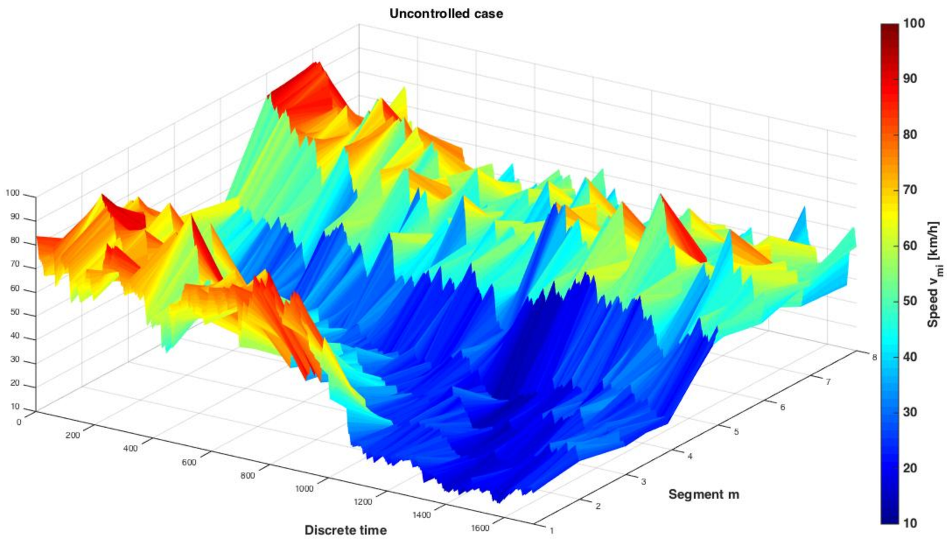


Figure 7.4.6: 3D mean speed plot, scenario 4, uncontrolled.

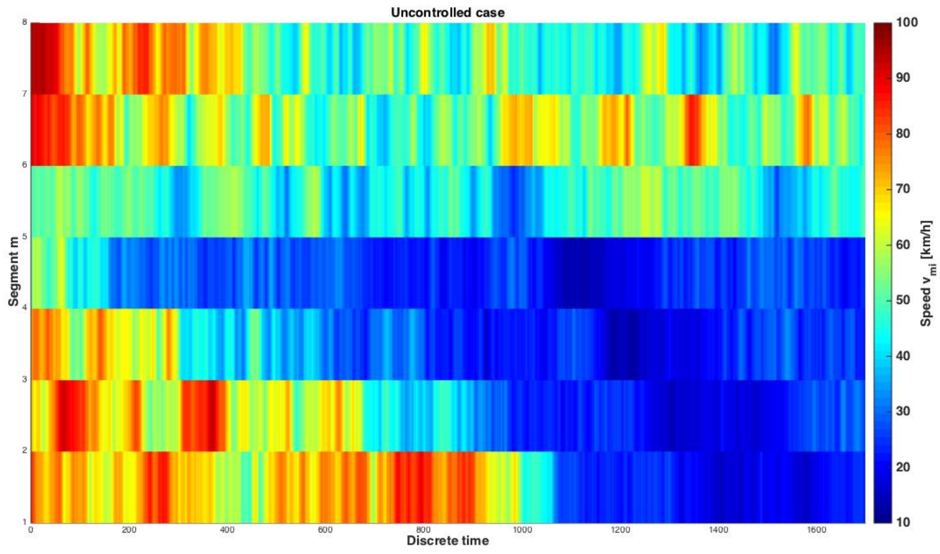


Figure 7.4.7: 2D view mean speed plot, scenario 4, uncontrolled.

When a controller is implemented, the mean speed is a little bit lower than in the uncontrolled case, which is indicated by the red areas in figure 7.4.6 and 7.4.7 in comparision to the orange areas in figure 7.4.9 and 7.4.8. In the congested area the speed is reduced to about 30-40 km/h and in the segments ahead of the congestion there are more light blue and green areas than in the uncontrolled scenario. To summarize, both the mean speed and the density was overall lower for the case where the controller were implemented. This indicates that the level of congestion can be decreased by controlling the speed limits.

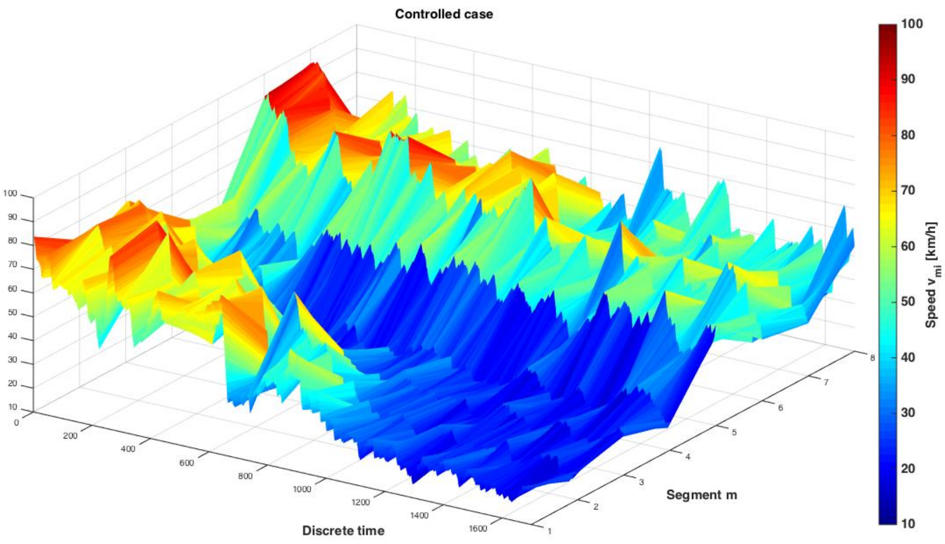


Figure 7.4.8: 3D mean speed plot, scenario 4, controlled.

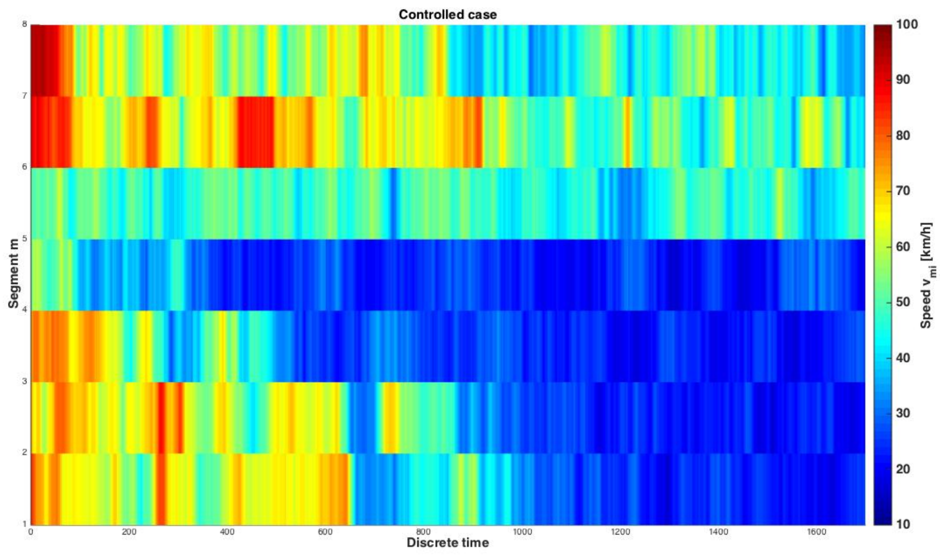


Figure 7.4.9: 2D view mean speed plot, scenario 4, controlled.

Figure 7.4.10 is displaying how the speed limit changes over time for some of the segments. It can be observed that the speed limits in segment 3 and segment 4 is similar until about time 660, and differs after that. Same goes for the speed limits in segment 5 and segment 6. In segment 6 the speed is consistent for a longer time period, while in segment 5 the limit varies. The reason why the speed limits are quite similar in segments close to each other is due to that if changes in on segment is made, the speed is also changed to the same limit in the previous segment. In METANET, the control variable could change the value for the speed limit during short periods of time (e.g. figure 7.3.4). Those fast changes does not exist in the figures below since the COM interface only allows the speed limit to change every 20th simulation second. This is more similar to how VSL would behave in the reality.

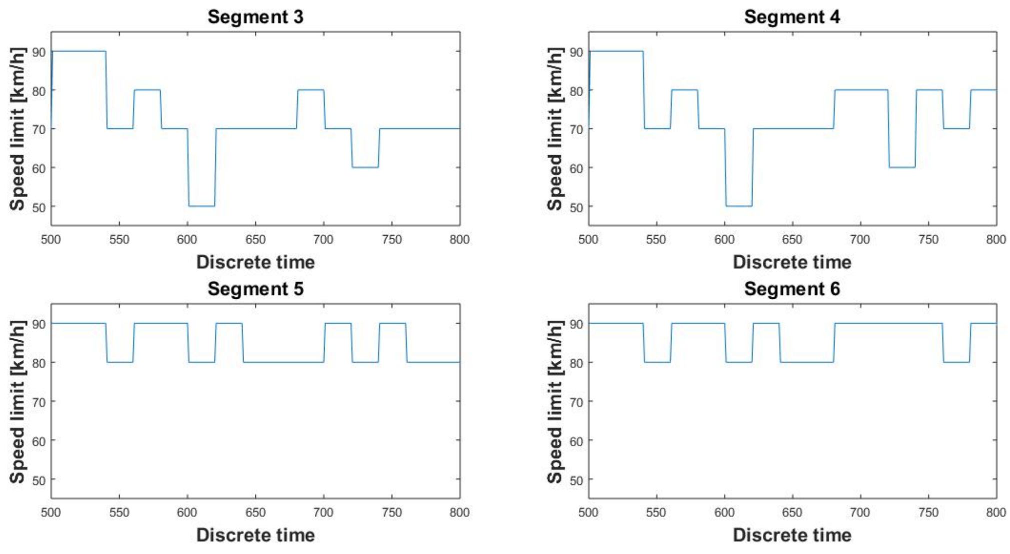


Figure 7.4.10: Change in speed limit over time, in segment 3-6

7.4.2 Scenario 5

When performing simulations for scenario 5, input values were chosen according to table 7.8. The inputs in scenario 5 were chosen with the aim to create a congestion due to a high inflow on the link as well as on from the ramp. In the last time interval the flow were decreased due to not creating a too severe congestion.

In figure 7.4.11 and figure 7.4.12 it can observed that the density is below critical in segment 6-7 with no great risk for congestion. The density in segment 5 is around or higher than the critical value during the whole simulation time, i.e. there can be congestion during some periods. This increased value can be due to the on-ramp connected to that segment. In the segments placed before the on-ramp, there is a higher level of congestion (~50-60 *veh/km/lane*). This congestion is propagating backwards and is caused by a combination of high inflow from the on-ramp and a high flow on the main link.

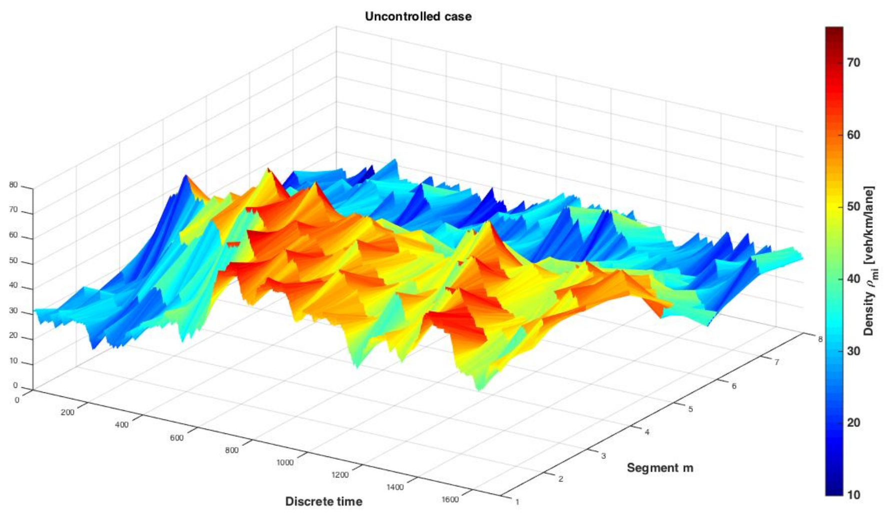


Figure 7.4.11: 3D density plot, scenario 5, uncontrolled.

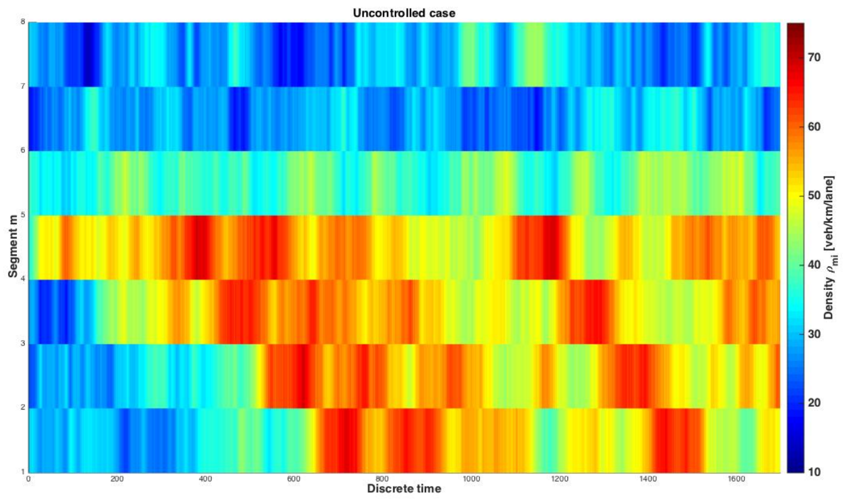


Figure 7.4.12: 2D view density plot, scenario 5, uncontrolled.

When VSL was implemented on scenario 5, the area of the red parts was reduced which can be observed in figure 7.4.13 and figure 7.4.14. Also, the highest peaks in density were lower than in the uncontrolled case (figure 7.4.12 and 7.4.11). The increased value in density is delayed in the case where VSL is applied ($\sim$ time 430), compared to the uncontrolled case ($\sim$ time 350). Further it can be observed that the density is higher, especially after the time 400, in the first four segments compared to the last three segments which is due to the same reason as in the uncontrolled case. Still, the density is lower during the congestion phase for the controlled case than for the uncontrolled case.

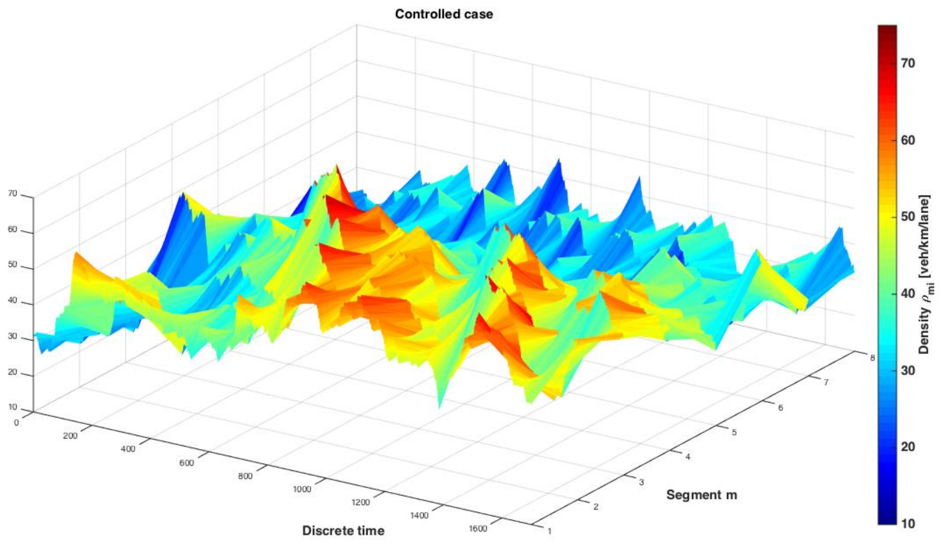


Figure 7.4.13: 3D density plot, scenario 5, controlled.

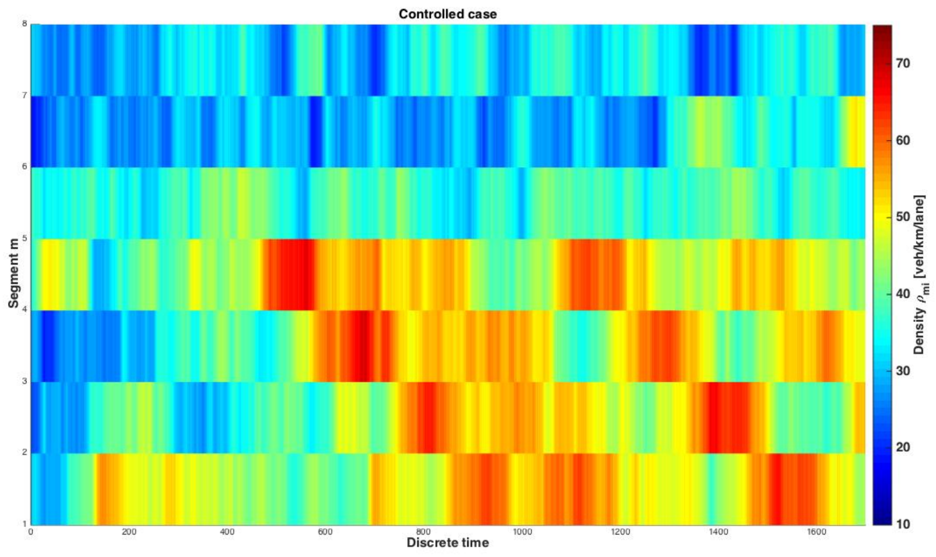


Figure 7.4.14: 2D view density plot, scenario 5, controlled.

Figure 7.4.15 indicates similar results as figure 7.4.5 but for segment 4, which is the segment downstream from where the on-ramp is connected. Here the density is overall higher and the peaks are higher as well compared to figure 7.4.5. In this case, the red curve is overlapping or is below the blue curve during almost the whole simulation. This indicates that the traffic flow has been improved by implementing the controller, since the density is overall lower during the controlled case. The biggest difference between the uncontrolled and the controlled case can be seen in the beginning and in the end of the simulation.

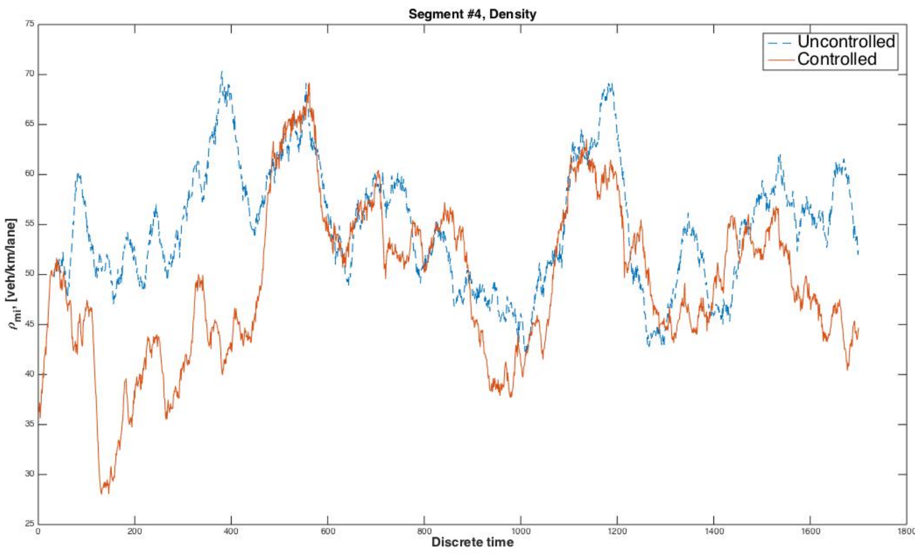


Figure 7.4.15: Density plot, scenario 5, segment 4.

Figure 7.4.16 and 7.4.17 indicates that the mean speed has decreased in segment 1-4 during the same period of time that the density increased in figure 7.4.11 and 7.4.12. The mean speed in segment 6-7 is higher, with some exceptions where the mean speed has been decreased during some periods of time. Segment 5 is affected by the connected on-ramp, as seen in figure 7.4.17 the mean speed is approximately 40-60 km/h for this segment during the whole simulation. The region where improvements is needed the most is the first four segments where the mean speed goes down to $\sim$30 km/h.

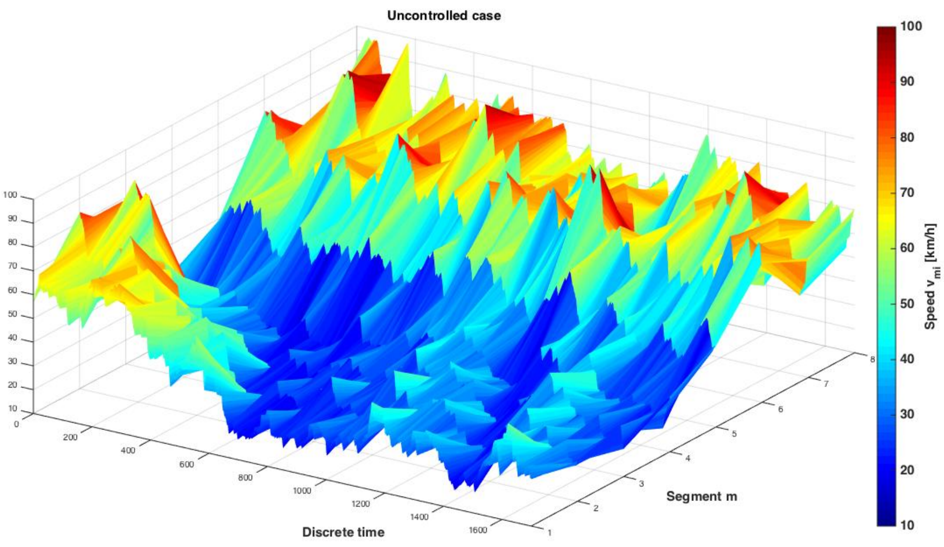


Figure 7.4.16: 3D mean speed plot, scenario 5, uncontrolled.

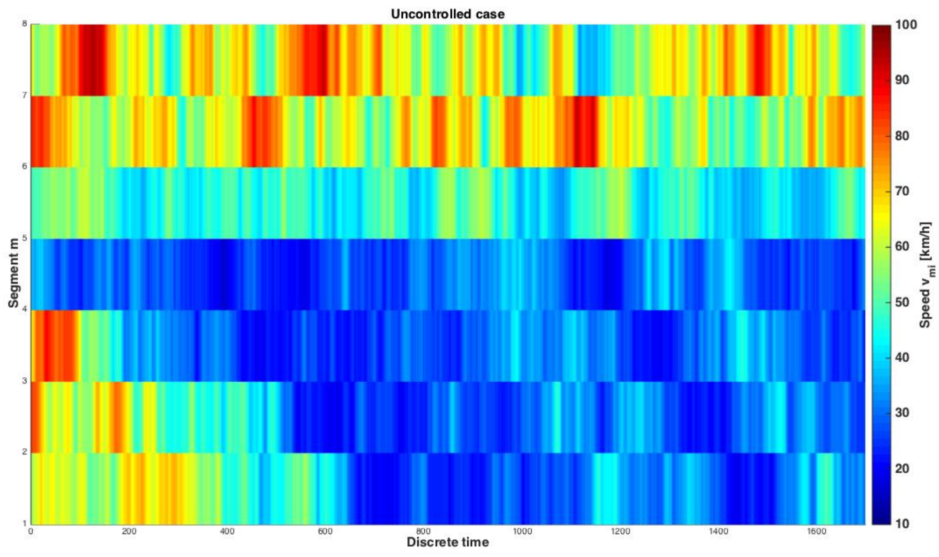


Figure 7.4.17: 2D view mean speed plot, scenario 5, uncontrolled.

When the controller is implemented (figure 7.4.18 and 7.4.19), the variation in mean speed does not differ as much throughout the link as in the uncontrolled case. During the congestion phase, the mean speed is higher when VSL is implemented. This indicates that the level of congestion has been reduced compared to the uncontrolled case. In the segments where the speed was high in the uncontrolled case, the mean speed is reduced for the controlled case due to the activation of the controller. Overall, the traffic flow is improved since the speed and density does not varies as much as in the uncontrolled case. In reality, this makes it possible for the drivers to assume a more constant speed.

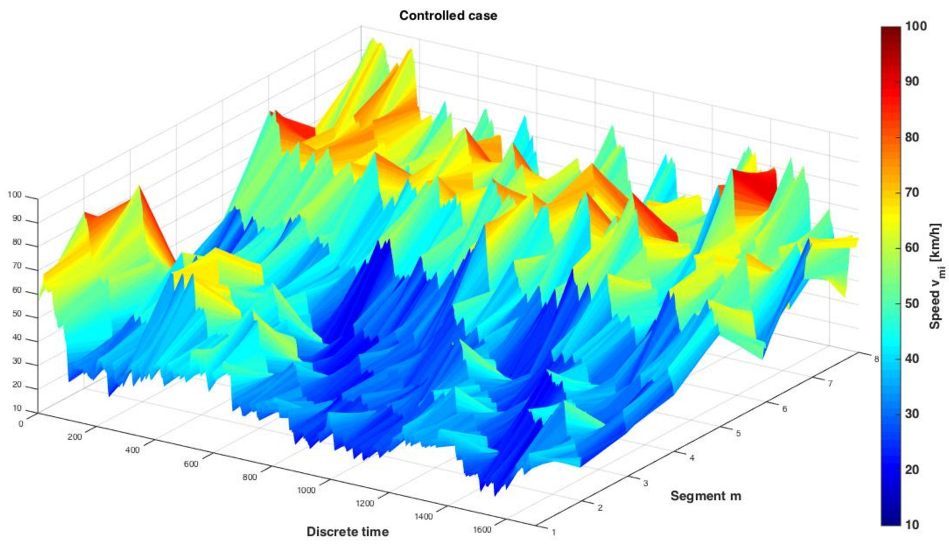


Figure 7.4.18: 3D mean speed plot, scenario 5, controlled.

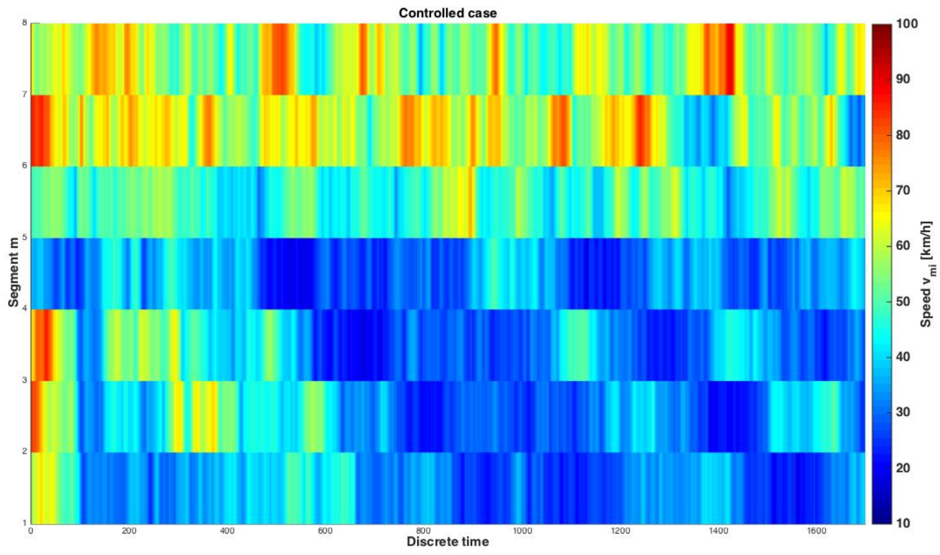


Figure 7.4.19: 2D view mean speed plot, scenario 5, controlled.

8 Discussion

In this chapter a review of the results are presented and explained. Both the results from the METANET model and the results from the VISSIM simulations performed through the COM interface are reviewed and compared.

The calibration and validation is important to make a realistic model because of the fact that METANET as well as our controller is depending on the design parameters. A lot of testing with different approaches and values on boundary conditions were implemented to try to optimize the fit and design parameters and still maintain a realistic scenario. The problem with the fitting was especially located in the congested regime. This can be related to the fact that the simulation results differ the most when the network is congested, which is reflected in a worsen fit between METANET and VISSIM data. The first selected distance of 7 km of the highway E6/E20 also made the fit more difficult. The reason for this was when a first attempt to fit METANET to a simulation with more than 6 segments, the results were inadequate for the segments in the middle. This is due to that we have set boundary conditions in the first and last segment and this is the values that the remaining segments depends on. So, the flow depends on the value for the flow in the previous segments. As we move forward in the link the deviation between the model and the data increases. This is not always a problem but in this specific simulation it was and therefore the length of the link were chosen to 3 km instead which eliminated the problem. Also when selecting boundary conditions by ourselves, that is presented in section 7.3, this was not a problem and a distance of 7 km could be applied.

An additional problem regarding the fitting emerged when adding the equations and the extra design parameter δ for the on-ramp. If the on-ramps were removed, the microscopic traffic model would be less realistic due to that all highways haves on-ramps in different sections. Therefore it was important to add an on-ramp and resemble a simplified model of E6/E20 in Mölndal but also to create a more general scenario that can be used for similar highways. On-ramp were prioritized due to their ability to create congestion. An off-ramp can also create a backwards propagating congestion, due to limited accessibility to the ramp during rush hours. However, simulations for a backwards propagating congestion were performed which can be compared to an off-ramp with a high flow that tries to enter the ramp.

The LQR controller is designed for a link with specific characteristics such as design parameters and speed limits. Whether the controller should improve the density and flow on other links with different length, on-/off-ramps and number of lanes is yet to be tested. The controller should be able to regulate other link networks as well, but not with the same success because of the simplification by excluding off-ramps in the model. The model can be further developed and combined with off-ramps to make it more realistic. Minor changes in the linearization is also required such as changing the current speed limit, as well as minimizing the functions to get better suited design parameters when implementing the controller on a new road network.

Real time measurements from the road where the case study was performed were supposed to be implemented on the simulated road in VISSIM and in the METANET model. A problem was that data often were missing during several time intervals and frequently not measured and book-kept at all. Because of this, all the data was not implemented in the simulations, just the data from the time period when the traffic was most dense (15:00-18:00) were applied to emulate E6/E20 Mölndal during rush hours. For the rest of the time periods it was no point in implementing the data, since it was not near any critical values and would therefore not affect the traffic activity to that point where VSL control is applied. Instead different scenarios were tested to find an interesting scenario with the right ratio between congestion and free flow. The scenarios had a higher level of congestion than the real time measurements from E6/E20 Mölndal has and since the controller improved the traffic flow for those scenarios, it will probably have a positive effect and lower the level of congestion in Mölndal as well. The controller were tested on the different scenarios to strengthen and validate the controller. It is of importance that the controller will give positive results on different types of congestion and different values on the density level for the controller being worth implementing in reality.

In all three scenarios for METANET (section 7.2 and 7.3) the VSL showed great improvements on the traffic flow. However, those results can be a bit unrealistic due to the simplification of the model by not using Cremer's model and instead assuming that all the vehicles comply to the speed limit. The improvements with the use of VSL would probably not be as good as presented in section 7.2.3 if Cremer's model would be applied. In real life all drivers do not follow the speed limit and therefore the effect of VSL and speed limits in general is not as successful as wished. We think that the results would still show an improvement if Cremer's models should be implemented because of the visually significant improvements showed in the figures in section 7.2.3 with the use of the controller. This is also a possible explanation to why the results from VISSIM , with the simulation controlled through the COM interface, were not as positive as the results from METANET, due to the drivers compliance level taken into consideration in the software. Overall, the implementation of VSL through the COM interface in VISSIM, decreased the density in the two scenarios in section 7.4. The decrement in density is visible, but not as distinguishable as the decrement for the scenarios in METANET. The result from 7.4 also gave a more homogeneous mean speed and resulted in a higher mean speed during the congestion phase in general for the controlled case than for the uncontrolled case.

Another result seen in section 7.2.2 and 7.2.3 is the fast changing behavior for the controller at some periods. The speed limit is displaying a fluctuating output even though a filter is applied to the signal. That is due to the sampling rate being high and the model responding too quickly to the fast changes in speed limits whenever the density ρ is near the critical density ρ_{cr}. In reality this problem will probably not be of importance due to a lower sampling rate of measurements and the density taken from measurements not responding so quickly to the change in speed limit. In the results regarding the two scenarios for the COM interface, the sampling time for the measurements were 20 seconds which showed a more smooth control. In reality the sampling rate for measurements could and should probably be even lower.

All simulations did fall below 2000 *simulation seconds*. It can be questioned whether 2000 *simulation seconds* is enough time to analyze the link and draw verified conclusions on the results or if the simulated time should be longer to get more realistic results. One problem with longer simulation time is the fact that it is very computationally heavy and time consuming.

9 Conclusion

Performing simulations and analyzing the use of VMS and VSL in Gothenburg today, has given a clear indication that there are room for improvements for the currently used control strategy. The given proposal is a more advanced control design of VSL than the current control design in Gothenburg. This proposal uses already existing, up and operating VMS signs in Gothenburg. Hence, the focus is not on changing already existing signs and sensors but instead creating a more advanced control algorithm. Since the present VSL control algorithm in the Mölndal E6/E20 area only reduces the speed if the speed experienced in the left lane drops below a certain value, the VSL signs can be switched on too late. Our proposal consist of a more advanced control design which not only take the speed into consideration when lowering the speed, but also the density with the aim at improving the flow before the congestion gets too severe. When speed is the only variable taken into consideration to decrease the speed limit, it is possible that a congestion already has been developed since the reduced speed is a result of a higher inflow of vehicles and limited accessibility to the link. The developed controller design take the density, speed but also the flow into consideration and results in a more homogeneous traffic flow on the highway.

For future research, off-ramps can be implemented in both the microscopic and macroscopic models to emulate a more realistic scenario of the highway. By doing this, the model will be more general and it will be easier to apply it on other links than the one in the selected case study. Another recommendation is to analyze how the decrement in speed should affect the following segments, i.e. how the speed limit for each segment should depend on each other. When simulating with the COM interface, the link was only 4 km and the allowed speed is 90 km/h. Because of this, a reduced speed in one segment will most likely lead to reduce speed in all of the segments behind. If the designed controller was used in reality, the link would be longer than 4 km and a more advanced control logic would be requiered. As discussed above, it is unrealistic that all drivers follow the speed limits. By implementing Cremer's model in the METANET model the drivers compliance level is taken into consideration and the controller design would probably be more realistic. To get an even better fit between the METANET and VISSIM data, each segment could be tried to fit individually and later calculate an average for the whole section. This might result in a better fit for the whole link and thereby also lead to more accurate design parameters. Possible improvements on the environmental perspective would also be an interesting further research. With decreasing emissions, due to more homogeneous mean speed and less idle driving this could be a positive consequence of the VSL and a possible area for future research.

To summarize, the use of VSL and VMS have many advantages and is a very useful tool to use in control of different traffic situations. The subject has a lot of potential to be explored and further developed.

References

Abdel-Aty, M. and Yu, R. (2014). Integrating safety in developing a variable speed limits system. *UTC National Center for Transportation Systems Productivity and Management.*

Bansal, P. (2013). Development of matlab-vissim interface for green time optimazion, ramp metering and variable speed limit signs. Internship report, Indian Institute of Technology Dehli.

Bordons, C. and Camacho, E. F. (1998). A generalized predictive controller for a wide class of industrial processes. *Control Systems Technology, IEEE Transactions on*, 6(3):372–387.

Cremer, M. and Papageorgiou, M. (1981). Parameter identification for a traffic flow model. *Automatica*, 17(6):837–843.

David Dixon (2013). Active traffic management. http://www.geograph.org.uk/photo/3709375. [Online; accessed April 15 , 2016].

Google maps (2016). Göteborg. https://www.google.se/maps/place/Göteborg/@57.7016455, 11.613511,10z/data=!3m1!4b1!4m5!3m4!1s0x464f8e67966c073f:0x4019078290e7c40!8m2!3d57. 70887!4d11.97456. [Online; accessed May 13 , 2016].

Hegyi, A. (2004). *Model predictive control for integrating traffic control measures.* TU Delft, Delft University of Technology.

Hegyi, A., De Schutter, B., and Hellendoorn, J. (2005). Optimal coordination of variable speed limits to suppress shock waves. *Intelligent Transportation Systems, IEEE Transactions on*, 6(1):102–112.

Highways England (2013). Controlled motorway. https://www.flickr.com/photos/highwaysagency/6280791097/in/album-72157627853500055/. [Online; accessed April 15 , 2016].

Messner, A. and Papageorgiou, M. (1990). Metanet: A macroscopic simulation program for motorway networks. *Traffic Engineering & Control*, 31(8-9):466–470.

Nasir, A. N. K., Ahmad, M. A., and Rahmat, M. F. (2008). Performance comparison between lqr and pid controllers for an inverted pendulum system. In *International Conference on Power Control and Optimization: Innovation in Power Control for Optimal Industry*, volume 1052, pages 124–128. AIP Publishing.

Nissan, A. and Koutsopoulosb, H. N. (2011). Evaluation of the impact of advisory variable speed limits on motorway capacity and level of service. *Procedia-Social and Behavioral Sciences*, 16:100–109.

Nygårdhs, S. and Helmers, G. (2007). Vms-variable message signs: A literature review. *VTI rapport*, 570A.

Papageorgiou, M., Papamichail, I., Messmer, A., and Wang, Y. (2010). Traffic simulation with metanet. In *Fundamentals of traffic simulation*, pages 399–430. Springer.

Prasad, L. B., Tyagi, B., and Gupta, H. O. (2012). Modelling and simulation for optimal control of nonlinear inverted pendulum dynamical system using pid controller and lqr. In *Modelling Symposium (AMS), 2012 Sixth Asia*, pages 138–143. IEEE.

Ronchi, E., Nilsson, D., Modig, H., and Walter, A. L. (2016). Variable message signs for road tunnel emergency evacuations. *Applied ergonomics*, 52:253–264.

Vägverket (2016). Trafikstockningar. http://wwwproj.arena.vv.se/rtdb/. [Online; accessed April 15 , 2016].

Appendices

A METANET

```
    This code represents the METANET model for traffic flow
dsl = 0;
d_o = 1000;
link_length = 9;
lambda_m = 3;
v_fm = 125;
rho_cr = 31.1541;
delta = 0.0400;
tau = 0.006;                          % Time constant
nu = 62.0642;                         % Anticipation constant
alpha_m = 1.1867;                     % Nonlinear relation between mean speed and den
kappa = 20;
K=1000;
L_m = .5;                             % Length of the segment
N_m = link_length/L_m;                % Number of segments
T = 10/3600;                          % Period
%u = 0.73;                            % Control variabel

rho_mi = (rho_cr-10)*ones(N_m,K);     % Density (veh/km/lane)
V = ones(N_m,K);                      % V without dynamic speed limits
%V_dsl1 = ones(N_m,K);                % V dynamic speed limit 1

u = zeros(N_m,K);
r_o = ones(1,K);                    % No rampmetering
q_o_hat = ones(1,K);
q_o_1_hat = ones(1,K);
q_o_2_hat = ones(1,K);
if dsl == 1
    [k_lqr, vv] = linearization_dsl();
end
v_mi = 110*ones(N_m,K);             % Mean speed (km/h)
q_mi = 3000*ones(N_m,K);            % Flow (veh/h)
w_o = ones(1,K);                    % queue length
q_o = ones(1,K);                    % outflow
rho_max = 180;                      % maximal density
Q_o = 7000;                         % maximal capacity
r=1000;

bc_q = 3000*ones(1,K);              % Boundary conditions
bc_v = 110*ones(1,K);
bc_r = ones(1,K);

for j=1:K
    if (j<450 || j>700)
        bc_q(j) = 3000+10*rand-10*rand;
        bc_v(j) = 110+10*rand-10*rand;
        rho_mi(N_m,j) = rho_cr-5 +10*rand-10*rand;
    else
        bc_q(j) = 3000+10*rand-10*rand;
        bc_v(j) = 110+10*rand-10*rand;
        rho_mi(N_m,j) = rho_cr+15 +10*rand-10*rand;
        %u(j) = 0.75;
    end
end


counter = 0;
% tau2=0.1;
% Ts=10/3600;
% xf=zeros(N_m,K);
% uf=zeros(N_m,K);

for k=1:K-1                          % For every time step
    for i=1:N_m-1                    % For every segment i the motorway link
        if i == 1
            q_mi(i,k) = bc_q(k);
            v_mi(i,k) = bc_v(k);
            rho_mi(i,k) = q_mi(i,k)./(v_mi(i,k)*lambda_m);
        elseif i < N_m
           if dsl == 1
           u(i,k)=-k_lqr*[rho_mi(i,k)-rho_cr;v_mi(i,k)-vv(2);w_o(k)-0]+110;
%              [a,b,c,d]=tf2ss(1/tau2,[1 1/tau2]);
%              [sysd]=c2d(ss(a,b,c,d),Ts);
%              [ad,bd,cd,dd]=ssdata(sysd);
%              xf(i,k+1)=ad*xf(i,k)+bd*uf(i,k);
%              u(i,k)=cd*xf(i,k)+dd*uf(i,k);
            if u(i,k) >= 120
                u(i,k) = 120;
            elseif u(i,k) < 120 && u(i,k) >= 90
                u(i,k) = 90;
            elseif u(i,k) < 90 && u(i,k) >= 70
                u(i,k) = 70;
            elseif u(i,k) < 70 && u(i,k) >= 50
                u(i,k) = 50;
            else
                u(i,k) = 30;
            end
            end
            q_mi(i,k) = rho_mi(i,k)*v_mi(i,k)*lambda_m;
            if i == 5
                rho_mi(i,k+1) = rho_mi(i,k) + (T/(L_m*lambda_m))*...
                    (q_mi(i-1,k) - q_mi(i,k)+r);
                q_o_1_hat(k) = d_o + w_o(k)/T;
                q_o_2_hat(k) = Q_o*min(1,((rho_max-rho_mi(i,k))/(rho_max...
                    - rho_cr)));
                q_o_hat(k) = min(q_o_1_hat(k),q_o_2_hat(k));
                q_o(k) = r_o(k)*q_o_hat(k);
                w_o(k+1) = w_o(k) + T*(d_o-q_o(k));
                if dsl == 1
                    if rho_mi(i,k) > rho_cr
                        V(i,k) = u(i,k);
                        counter = counter +1;
                     else
                         V(i,k) = v_fm * exp((-1/alpha_m)*...
                             (rho_mi(i,k)/rho_cr)^alpha_m);
                    end
                else
                    V(i,k) = v_fm * exp((-1/alpha_m)*...
                        (rho_mi(i,k)/rho_cr)^alpha_m);
                end
                v_mi(i,k+1) = v_mi(i,k) + ...
                    (T/tau)*(V(i,k)-v_mi(i,k)) + ...
                    (T/L_m)*(v_mi(i-1,k)-v_mi(i,k))* ...
                    v_mi(i,k) - (nu*T/(tau*L_m))*((rho_mi(i+1,k)...
                    -rho_mi(i,k))/(rho_mi(i,k)+kappa)) - ...
                    (delta*T*q_o(k)*v_mi(i,k))/...
                    (L_m*lambda_m*(rho_mi(i,k)+kappa));
            else
                rho_mi(i,k+1) = rho_mi(i,k) + ...
                   (T/(L_m*lambda_m))*(q_mi(i-1,k) - q_mi(i,k)) +rand-rand;
                if dsl == 1
                    if rho_mi(i,k) > rho_cr
                        V(i,k) = u(i,k);
                        counter = counter +1;
                    else
                       V(i,k) = v_fm * exp((-1/alpha_m)*...
                            (rho_mi(i,k)/rho_cr)^alpha_m);
                    end
                else
                    V(i,k) = v_fm * exp((-1/alpha_m)*...
                        (rho_mi(i,k)/rho_cr)^alpha_m);
                end
                v_mi(i,k+1) = v_mi(i,k) + (T/tau)*(V(i,k)-v_mi(i,k)) + ...
                (T/L_m)*(v_mi(i-1,k)-v_mi(i,k))* v_mi(i,k) - ...
                (nu*T/(tau*L_m))*((rho_mi(i+1,k)-rho_mi(i,k))/...
                (rho_mi(i,k)+kappa));
            end

            if v_mi(i,k+1) < 0
                v_mi(i,k+1) = 0;
            end

            if rho_mi(i,k+1) < 0
                rho_mi(i,k+1) = 0;
            end

            if k == (K-1)
                q_mi(i,k+1) = rho_mi(i,k+1)*v_mi(i,k+1)*lambda_m;
            end
%           else
%               rho_mi(i,k) = rho_mi(i-1,k);
        end
    end
end
if dsl == 1
    rhodsl = rho_mi;
    vdsl = v_mi;
    Vdsl = V;
    udsl = u;
end
```

B Load excel

```
global speedmatrix_1 speedmatrix_2 densitymatrix_1 densitymatrix_2

startvalue=300;
simulationtime=2000-300;
no_segments= 14;
no_links=2;
timestep=1;
matrix_all=zeros(5,no_links*no_segments*(simulationtime/timestep));
speedmatrix_1=zeros(simulationtime/timestep,no_segments);
densitymatrix_1=zeros(simulationtime/timestep,no_segments);
speedmatrix_2=zeros(simulationtime/timestep,no_segments);
densitymatrix_2=zeros(simulationtime/timestep,no_segments);
vissresult = xlsread('results_12.xlsx');

matrix_densityspeed_1= vissresult(1:23821-21,3:7);

for i=1:size(matrix_densityspeed_1,1) %segment,tid

       densitymatrix_1(matrix_densityspeed_1(i,1),matrix_densityspeed_1(i,2))=
       matrix_densityspeed_1(i,3);
       speedmatrix_1(matrix_densityspeed_1(i,1),matrix_densityspeed_1(i,2))=
       matrix_densityspeed_1(i,4);

end

matrix_densityspeed_2= vissresult(23821-21:47621-21,3:7);

for i=1:size(matrix_densityspeed_2,1) %segment,tid

       densitymatrix_2(matrix_densityspeed_2(i,1),matrix_densityspeed_2(i,2))=
       matrix_densityspeed_2(i,3);
       speedmatrix_2(matrix_densityspeed_2(i,1),matrix_densityspeed_2(i,2))=
       matrix_densityspeed_2(i,4);

end

speedmatrix_1 = speedmatrix_1';
speedmatrix_2 = speedmatrix_2';
densitymatrix_1 = densitymatrix_1';
densitymatrix_2 = densitymatrix_2';
```

C Minimization

```
global V g densitymatrix_1 densitymatrix_2 speedmatrix_1
global speedmatrix_2 densitymatrix speedmatrix
x0 = [120, 1.567, 30, 15/3600, 60, 40];
ub = [140, 2.1, 50, 0.0060, 80, 60];
lb = [90, 1.5, 10, 0.0040, 40, 10];


rhovis_mi=[densitymatrix_1;densitymatrix_1(end,:)];
vvis_mi=[speedmatrix_1;speedmatrix_1(end,:)];

options = optimoptions(@fmincon,'Display','iter');
[x,fval] = fmincon(@myfun,x0,[],[],[],[],lb,ub,[],options);
```

D Objective function

```
function g = myfun2( z )
global  speedmatrix_1 densitymatrix_1 flowmatrix_1 %speedmatrix_2

v_fm = z(1);%86;                    % Free sped on link m
alpha_m = z(2);                     % Nonlinear relation between mean speed and density
rho_cr = z(3);%37;                  % Critical density per lane of link m
tau = z(4);                         % Time constant
nu = z(5);                          % Anticipation constant
kappa = z(6);
delta = z(7);

% Variables for the chosen motorway link m
link_length = 3;
L_m = .5;                           % Length of the segment
lambda_m = 3;                       % Number of lanes in the segment
N_m = link_length/L_m;              % Number of segments
%N_m = 12;
K = 1700;                           % Time horizon
T = 1/3600;                        % Period
u = 0.73;                           % Control variabel
rho_mi = 30*ones(N_m,K);            % Density (veh/km/lane)
V = ones(N_m,K);                    % V without dynamic speed limits
V_dsl1 = ones(N_m,K);               % V dynamic speed limit 1
g=0;

r_o = ones(1,K);                    % No rampmetering
q_o_hat = ones(1,K);
q_o_1_hat = ones(1,K);
q_o_2_hat = ones(1,K);


v_mi = 60*ones(N_m,K);              % Mean speed (km/h)
q_mi = 1800*ones(N_m,K);            % Flow (veh/h)
w_o = ones(1,K);                    % queue length
q_o = ones(1,K);                    % outflow
d_o = 1000;                         % demand
rho_max = 180;                      % maximal density
Q_o = 7000;                         % maximal capacity
r=1000;


for j=1:K
    rho_mi(N_m,j) = densitymatrix_1(N_m,j);
    v_mi(N_m,j) = speedmatrix_1(N_m,j);
    q_mi(N_m,j) = flowmatrix_1(N_m,j);
    v_mi(1,j) = speedmatrix_1(1,j);
    q_mi(1,j) = flowmatrix_1(1,j);
    rho_mi(1,j) = densitymatrix_1(1,j);
end


for k=1:K-1                          % For every time step
    for i=2:N_m-1                    % For every segment i the motorway link
```

```
        q_mi(i,k) = rho_mi(i,k)*v_mi(i,k)*lambda_m;
        V(i,k) = v_fm * exp((-1/alpha_m)*(rho_mi(i,k)/rho_cr)^alpha_m);
        V_dsl1(i,k) = v_fm * u * exp(-(1+u)/4*(rho_mi(i,k)/rho_cr)^2);
        if i == 5
            rho_mi(i,k+1) = rho_mi(i,k) + (T/(L_m*lambda_m))*(q_mi(i-1,k)...
                - q_mi(i,k)+r);
            q_o_1_hat(k) = d_o + w_o(k)/T;
            q_o_2_hat(k) = Q_o*min(1,((rho_max - rho_mi(i,k))/...
                (rho_max - rho_cr)));
            q_o_hat(k) = min(q_o_1_hat(k),q_o_2_hat(k));
            q_o(k) = r_o(k)*q_o_hat(k);
            w_o(k+1) = w_o(k) + T*(d_o-q_o(k));
            V(i,k) = v_fm * exp((-1/alpha_m)*(rho_mi(i,k)/rho_cr)^alpha_m);
            V_dsl1(i,k) = v_fm * u * exp(-(1+u)/4*(rho_mi(i,k)/rho_cr)^2);
            v_mi(i,k+1) = v_mi(i,k) + (T/tau)*(V(i,k)-v_mi(i,k)) + ...
                (T/L_m)*(v_mi(i-1,k)-v_mi(i,k))* v_mi(i,k) - ...
                (nu*T/(tau*L_m))*((rho_mi(i+1,k)-...
                rho_mi(i,k))/(rho_mi(i,k)+kappa)) - ...
                (delta*T*q_o(k)*v_mi(i,k))/(L_m*lambda_m*(rho_mi(i,k)+kappa));
        else
            rho_mi(i,k+1) = rho_mi(i,k) + ...
                (T/(L_m*lambda_m))*(q_mi(i-1,k) - q_mi(i,k));
            V(i,k) = v_fm * exp((-1/alpha_m)*(rho_mi(i,k)/rho_cr)^alpha_m);
            V_dsl1(i,k) = v_fm * u * exp(-(1+u)/4*(rho_mi(i,k)/rho_cr)^2);
            v_mi(i,k+1) = v_mi(i,k) + (T/tau)*(V(i,k)-v_mi(i,k)) + ...
                (T/L_m)*(v_mi(i-1,k)-v_mi(i,k))* v_mi(i,k) - ...
                (nu*T/(tau*L_m))*((rho_mi(i+1,k)-rho_mi(i,k))/...
                (rho_mi(i,k)+kappa));
        end

        if v_mi(i,k+1) < 0
            v_mi(i,k+1) = 0;
        end

        if rho_mi(i,k+1) < 0
            rho_mi(i,k+1) = 0;
        end

        if k == (K-1)
            q_mi(i,k+1) = rho_mi(i,k+1)*v_mi(i,k+1)*lambda_m;
        end
        g = g+((rho_mi(i,k)-densitymatrix_1(i,k))^2+...
            (v_mi(i,k)-speedmatrix_1(i,k))^2);
    end
end
end
```

E Linearization

```
function [k_lqr, vv] = linearization()
global x

syms x1 x2 x3  u
                                    % x1= rho_m,i(k), x2= v_m,i(k),
                                    % x3= w_o(k), u=control signal
v_fm  = x(1);                       %design parameter
rho_cr = x(3);                      %design parameter
tau = x(4);                         %design parameter
nu = x(5);                          %design parameter
kappa = x(6);                       %design parameter
delta = x(7);                       %design parameter

L_m = 0.5;
T = 10/3600;                          %Time period
lambda_m=3;                           %number of lanes

density_prev=rho_cr;
rho_next=rho_cr;
d_o=1000;
r=d_o;

[vv] = fsolve(@steady_state, [100,110]);
    %calculate the stationary point of v
speed_prev=vv(1);

f1= x1 + T/(L_m*lambda_m)*(lambda_m*speed_prev*density_prev-(x1*x2*lambda_m));
f2 = x2 + (T/tau)*(v_fm*u*exp((-(1+u)/4)*(x1/rho_cr)^2)-x2)+
(T/L_m)*x2*(speed_prev - x2)-(nu*T/(tau*L_m))*((rho_next-x1)/(x1+kappa))-
(delta*T*r*x2)/(L_m*lambda_m*(x1+kappa));
f3 = x3 + T*(d_o-r);
                              % The system
                              % f1=rho_m,i(k+1)
                              % f2=v_m,i(k+1)
                              % f3=w_o(k+1)


A=jacobian([f1;f2; f3],[x1 x2 x3]);
    %Gives the Jacobian matrix of the system with respect to x1,x2 and x3.
B=jacobian([f1;f2; f3],u);
    % Gives the Jacobian matrix of the system with respect to u.

Ae=eval(subs(A, [x1 x2 x3 u], [rho_cr vv(2) 0 0.75]));
    %Evaluating the Jacobian matrix A around the stationary point
Be=eval(subs(B,[x1 x2 x3 u], [rho_cr vv(2) 0 0.75]));
    %Evaluating the Jacobian matrix B around the stationary point
Ae(3,3)=.9999;
    %Element (3,3)=1 is replaced by 0.9999 (within the unit circle)
    %to make the system stabile
Qe=diag([1000,.010,.010]);
    %Qe matrix ranking the variables, 1000 is the rank of the density,
     %0.010 is the rank of the speed and queue length.
Re=0.001;
```

```
    %Re is set to 0.01 to make the changes in limits in VSL more smooth
k_lqr=dlqr(Ae,Be,Qe,Re,[]);
    % gives the control vector
end
```

F Steady state

```
function f = steady_state(vv)
global x

v_fm = 117.1950;
alpha_m = 1.4170;
rho_cr = 42.3802;
tau = .0040;
nu = 80.000;
kappa = 27.8260;
delta = 0.0400;

L_m = 0.5;
u = 110;
T = 10/3600;
r=1000;
V = u;

f = [(rho_cr*T/L_m)*vv(1) - (rho_cr*T/L_m)*vv(2)+T/(L_m*3)*r;
     (T/tau)*V - (T/tau)*vv(2) + (T/L_m)*vv(1)*vv(2) -
     (T/L_m)*vv(2)^2-(delta*T*r*vv(2))/(L_m*3*(rho_cr+kappa))];

disp(vv)

end
```

G COM interface

```
global x
format compact;
vis=actxserver('VISSIM.vissim.700');

access_path = pwd;
vis.LoadNet([access_path '\molndal7.inpx']);
% loading the current vissim *.inp file given with the whole root
vis.LoadLayout([access_path '\molndal7.layx']);
% loading the current vissim *.ini file given with the whole root

sim=vis.simulation;
timestep=1;
sim.set('AttValue','SimRes', timestep);
vnet=vis.net;

vehins = vnet.VehicleInputs;
vehin_1 = vehins.ItemByKey(1);
vehin_2 = vehins.ItemByKey(2);

% Set vehicle inputs

vehin_1.set('AttValue', 'Volume(1)', 5000);
vehin_1.set('AttValue', 'Volume(2)', 5000);
vehin_1.set('AttValue', 'Volume(3)', 5000);

vehin_2.set('AttValue', 'Volume(1)', 1000);
vehin_2.set('AttValue', 'Volume(2)', 1000);
vehin_2.set('AttValue', 'Volume(3)', 1000);

period_time = 2000;

links=vnet.Links;
link1=links.ItemByKey(1);
link2=links.ItemByKey(2);

speeddec=vnet.DesSpeedDecisions;
% Store the speed decisions in a cell array
speeddec_a={1,27};
for i=1:27
speeddec_a{i} = speeddec.ItemByKey(i);
end
for i=1:27
speeddec_a{i}.set('AttValue','DesSpeedDistr(10)',90);
end
% Set the speed decisions on the on-ramp
speeddec_a{9}.set('AttValue','DesSpeedDistr(10)',40);
speeddec_a{18}.set('AttValue','DesSpeedDistr(10)',40);
speeddec_a{27}.set('AttValue','DesSpeedDistr(10)',40);

datapoints = vnet.DataCollectionMeasurements;
% Store the data collection points in a cell array
datapoints_a={1,8};
for i=1:8
```

```
datapoints_a{i} = datapoints.ItemByKey(i);
end

%%
speed1=0;
speed_j=0;
% Optimized parameters
x = [98.022, 1.5969, 31.1096, 0.004, 72.6446,16.3038, 0.0296];
[k_lqr, vv]=linearizationCOM();
queue_length=0;
v_fm=x(1);
rho_cr=x(3);
uf=zeros(8,2000);
tau2=0.1;
Ts=10/3600;
xf=zeros(8,2000);
speed_lastsegment = 90;
speed_limit_matrix=90*ones(8,100);

for i=1:(2000)
    % Run the simulation by single step
    sim.RunSingleStep;
    sim.get('AttValue', 'SimSec');
    sim.get('AttValue', 'SimPeriod');
    % Let 300 s ellaps so there are vehicles in the network
    % Check every 20th second
    if(i>=300) && mod(i,20) == 0
    for j=8:-1:1
        lastspeed_j=speed_j;
        %
        speed_j=datapoints_a{j}.get('AttValue','Speed(Current, Last, All)');
        veh_j=datapoints_a{j}.get('AttValue','Vehs(Current, Last, All)');

        if isnan(speed_j)
            speed_j=lastspeed_j;
        end
        if isnan(veh_j)
            veh_j=0;
        end
        % Calculate the flow
        flow_j=veh_j*360;
        density_j=double(flow_j)/speed_j;
        % Calculate the contol variable u
        u(j,i) = - k_lqr*[density_j-rho_cr; speed_j-vv(2); queue_length-0]+90

        speed_vsl = u(j,i)
        speed_vsl = round(speed_vsl,-1);

        bool = 0;
        if speed_lastsegment < speed_vsl
           speed_vsl=speed_lastsegment;
           bool = 1;
        end
        disp(speed_vsl)
```

```
% No VSL contol
 if density_j<rho_cr && bool == 0
     speeddec_a{j}.set('AttValue','DesSpeedDistr(10)',90);
     speeddec_a{j+9}.set('AttValue','DesSpeedDistr(10)',90);
     speeddec_a{j+18}.set('AttValue','DesSpeedDistr(10)',90);
     speed_lastsegment=90;
 else
% VSL control
     hej(j,i)=u(j,i);
     if(speed_vsl <40)
         speeddec_a{j}.set('AttValue','DesSpeedDistr(10)',30);
         speeddec_a{j+9}.set('AttValue','DesSpeedDistr(10)',30);
         speeddec_a{j+18}.set('AttValue','DesSpeedDistr(10)',30);
         speed_lastsegment=30;
         speed_limit_matrix(j,i/20)=30;
     end
     if(speed_vsl >= 40 && speed_vsl <50)
         speeddec_a{j}.set('AttValue','DesSpeedDistr(10)',40);
         speeddec_a{+9}.set('AttValue','DesSpeedDistr(10)',40);
         speeddec_a{j+18}.set('AttValue','DesSpeedDistr(10)',40);
         speed_lastsegment=40;
         speed_limit_matrix(j,i/20)=40;
     end
     if(speed_vsl >= 50 && speed_vsl <60)
         speeddec_a{j}.set('AttValue','DesSpeedDistr(10)',50);
         speeddec_a{j+9}.set('AttValue','DesSpeedDistr(10)',50);
         speeddec_a{j+18}.set('AttValue','DesSpeedDistr(10)',50);
         speed_lastsegment=50;
         speed_limit_matrix(j,i/20)=50;
     end
     if(speed_vsl >= 60 && speed_vsl <70)
         speeddec_a{j}.set('AttValue','DesSpeedDistr(10)',60);
         speeddec_a{j+9}.set('AttValue','DesSpeedDistr(10)',60);
         speeddec_a{j+18}.set('AttValue','DesSpeedDistr(10)',60);
         speed_lastsegment=60;
         speed_limit_matrix(j,i/20)=60;
     end
     if(speed_vsl >= 70 && speed_vsl <80)
         speeddec_a{j}.set('AttValue','DesSpeedDistr(10)',70);
         speeddec_a{j+9}.set('AttValue','DesSpeedDistr(10)',70);
         speeddec_a{j+18}.set('AttValue','DesSpeedDistr(10)',70);
         speed_lastsegment=70;
         speed_limit_matrix(j,i/20)=70;
     end
     if(speed_vsl >= 80 && speed_vsl <90)
         speeddec_a{j}.set('AttValue','DesSpeedDistr(10)',80);
         speeddec_a{j+9}.set('AttValue','DesSpeedDistr(10)',80);
         speeddec_a{j+18}.set('AttValue','DesSpeedDistr(10)',80);
         speed_lastsegment=80;
         speed_limit_matrix(j,i/20)=80;
     end
     if(speed_vsl >= 90 && speed_vsl <100)
         speeddec_a{j}.set('AttValue','DesSpeedDistr(10)',90);
         speeddec_a{j+9}.set('AttValue','DesSpeedDistr(10)',90);
         speeddec_a{j+18}.set('AttValue','DesSpeedDistr(10)',90);
```

```
                speed_lastsegment=90;
            end
            if(speed_vsl >= 100)
                speeddec_a{j}.set('AttValue','DesSpeedDistr(10)',90);
                speeddec_a{j+9}.set('AttValue','DesSpeedDistr(10)',90);
                speeddec_a{j+18}.set('AttValue','DesSpeedDistr(10)',90);
                speed_lastsegment=90;
            end
        end
        if j==1
            speed_lastsegment = 90;
        end
    end
    end
end

vis.release;
disp('The End');
```