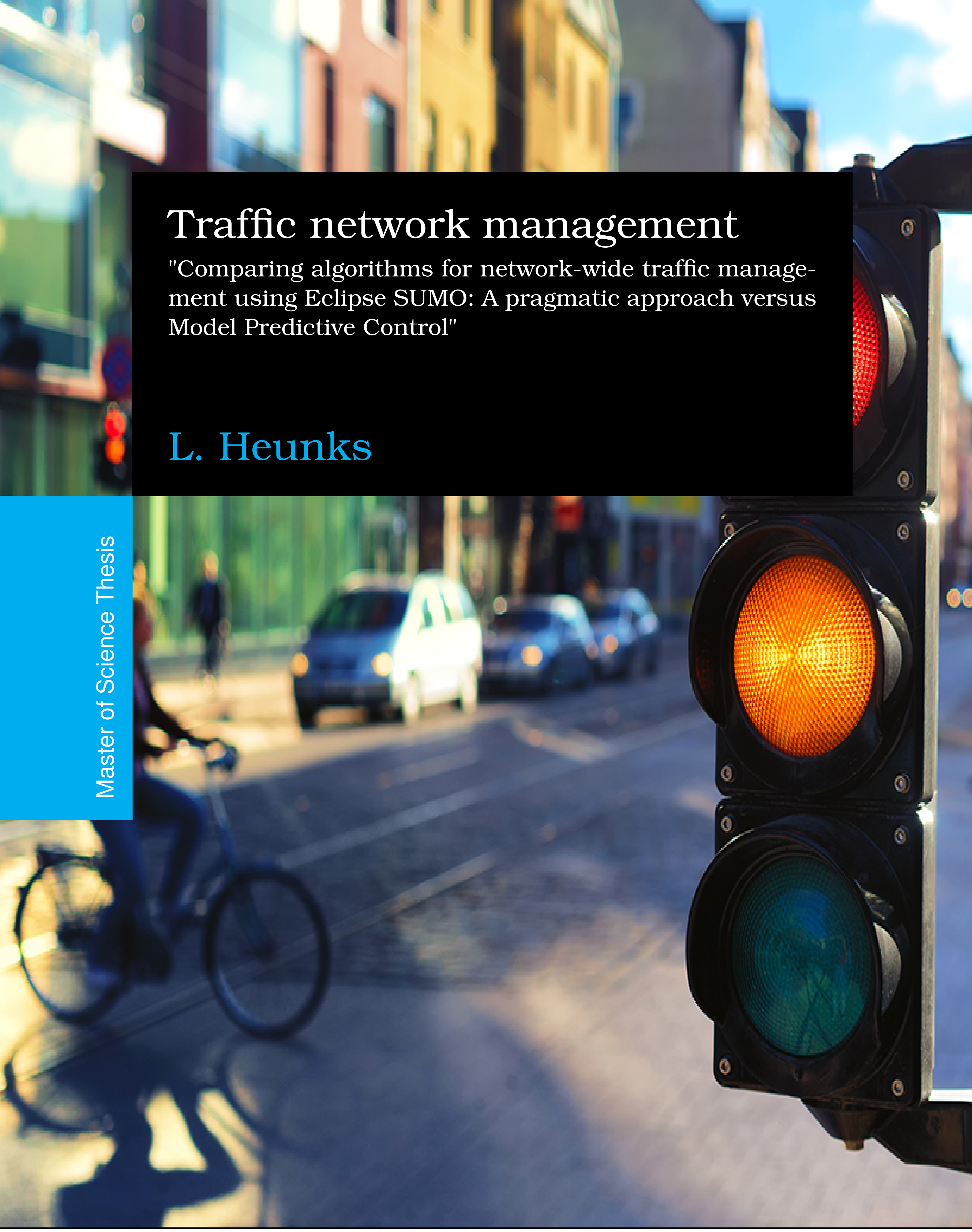


Traffic network management

"Comparing algorithms for network-wide traffic management using Eclipse SUMO: A pragmatic approach versus Model Predictive Control"

L. Heunks

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Traffic network management

"Comparing algorithms for network-wide traffic management using Eclipse SUMO: A pragmatic approach versus Model Predictive Control"

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Systems and Control at Delft University of Technology

L. Heunks

May 9, 2022

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported by Technolution. Their cooperation is hereby gratefully acknowledged.

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
DELFT CENTER FOR SYSTEMS AND CONTROL (DCSC)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

TRAFFIC NETWORK MANAGEMENT

by

L. HEUNKS

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE SYSTEMS AND CONTROL

Dated: May 9, 2022

Supervisor(s):

Dr.Ing. S. Grammatico

Ir. T. Van Bakel

Reader(s):

Dr. S.D. Gonçalves Melo Pequito

Dr. A. Dabiri

Abstract

The need for smart traffic control has grown over the last years. Initiated by an increased amount of traffic. Network-wide traffic control is becoming a more interesting field for traffic control. Mainly because computer power has increased and optimisation techniques improved. Network-wide traffic aims to improve the overall traffic state by looking at the entire problem instead of sub-problems. Besides improving traffic conditions, network-wide traffic control could support road operators in simplifying their work by taking over some tasks and keeping track of the situation.

For this research, we compare a pragmatic, user-friendly and transparent control method versus a Model Predictive Control (MPC) approach. For the MPC controller, the second-order macroscopic METANET model is chosen. The METANET model describes a network as a directed graph. We test both controllers on two small scale freeway traffic networks. The control measures that are implemented are ramp-metering and rerouting. The simulations are done in a SUMO environment. The key performance index (KPI) used for comparison is Total Time Spend (TTS). The resulting optimisation problem is a Mixed Non-linear Integer Problem (MINLP). This problem is solved with a heuristic method by a Genetic Algorithm (GA).

Simulations for the two networks in a demand scenario around critical density are analysed over 20 iterations. The results prove the potential of both algorithms since both improved the TTS significantly. The NM excels in ease of implementation and ease of understanding for non-experts. While the MPC outperforms the NM in TTS reduction, it is harder to configure and understand for non-experts. The MPC is successfully tested on its capability to prevent undesired behaviour from happening by adding penalties to the objective function. For future research, larger networks need to be investigated, with a focus on simplifying the resulting optimisation problem. It is expected that a piecewise affine approximation is a promising method.

Table of Contents

List of Figures

List of Tables

Preface

When the last exams of my master's approached, we were in the middle of the COVID-19 pandemic. I was sitting at home at my desk day after day. The view left of me was a busy intersection in the city centre of Rotterdam. Wandering in thoughts, my attention often went to this intersection. When this situation continued, I realised how the complexity and dynamics of traffic triggered my interest.

When I was connected to my supervisor Dr. S.D. Gonçalves Melo PequitoDr. Ing. S. Grammatico and there was a possible graduation project on traffic network management at Technolution, my attention was directly aroused. Within a few weeks, the agreements were set and I was ready to go.

The first months were enjoyable. I met many people from Technolution and experts from the field. These experts were from municipalities and companies who were partners with Technolution. Afterwards, I had to finish my literature study. Less enjoyable in the last weeks though useful and I got to know the practical and scientific field during this entire study.

During the practical part of this research, I learnt a lot and had a lot of useful discussions with my supervisors. The research and thesis continuously improved during the process, despite some natural and unavoidable setbacks. I am happy with the final results and the gained knowledge during the process. I would always advise starting as soon as possible by building a minimal viable product (MVP) for all the algorithms or variables you want to test. Try to not lose yourself in a single focus or configuration. One of the things I value is that the framework I built and the work that has been done, rises extra questions and ideas one could and want to research further.

Delft, University of Technology — L. Heunks
May 9, 2022

Acknowledgements

I like to thank my supervisor Sergio Grammatico from the TU Delft and my supervisor Tijs van Bakel from Technolution for their assistance during the writing and mainly during the research itself of this thesis. Besides, I am thanking Technolution for the support in general. Including all the other valuable moments with their employees like Paul van Koningsbruggen, Edwin Mein and William Meijer.

Delft, University of Technology
May 9, 2022

L. Heunks

Chapter 1

Introduction

This thesis paper focuses on network-wide traffic management via a case study of two different algorithms in different circumstances. This chapter introduces the reader to the topic of traffic management in section 1-1, the relevance of network-wide traffic management in section 1-2, the use-case and the purpose of this survey in section 1-3 and the shortcomings of current solutions.

1-1 Description of practice

Road traffic is a complex phenomenon. There are many different participants. Besides, interactions continuously take place. Road users can make multiple choices while participating in the traffic. To name a few: which route to take, which lane to use or what speed to drive. Furthermore, there are interactions between humans and vehicles and among the different road users. To control this inherent complexity, there are basic traffic rules that one learns from parents, during school and driving lessons. However, there is more that makes traffic complex, namely all the different traffic control systems and the people that manage these systems. These aspects cause extra interactions between the traffic control systems with road users and operators. All of these aspects contribute to the total complexity of traffic.

Within the field of traffic control, traffic rules are known as the basics. One level higher are the traffic control systems, which operate independently like traffic lights. Currently, most traffic lights already have some intelligence instead of simply doing their cycle of green and red time. At last, there is traffic management at the highest level. Traffic management is about regulating traffic with smart combinations of traffic control systems. Road operators can adjust traffic light schemes or make use of systems like variable speed limit signs and variable message signs.

In the practice of traffic management, there are many different organisations in the Netherlands. The road network consists of provincial roads, urban roads and rural roads. These are respectively managed by the province, the municipality and the central government called 'Rijkswaterstaat'. All these organisations play a role in regulating traffic. Each has its traffic centres, for an impression see figure Figure 1-1.

Figure 1-1: Traffic centre of 'Rijkswaterstaat'. [1]

In these centres, people monitor and manage the traffic 24 hours a day, 7 days a week. People know what to do for planned situations and they are prepared for unforeseen circumstances. For many situations, they have prepared scenarios on how to deal with the situation. Those scenarios are about synchronising multiple actuators like traffic lights, dynamic route information panels and (portable) variable message signs. All this is done in collaboration with the authorities of interest. By combining and activating these actuators traffic centres can regulate and influence traffic. The most important and rigorous scenarios are checked and signed by all involved authorities before they are used in practice. In figure 1-2 one can find an example for region Amsterdam in case of maintenance.

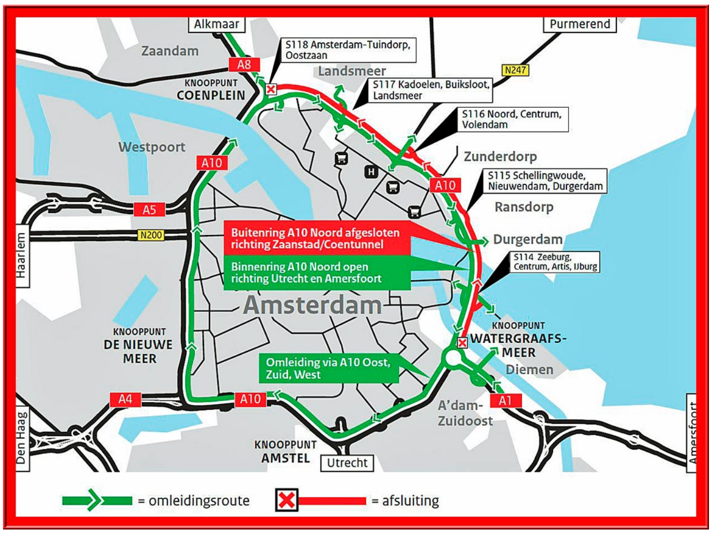


Figure 1-2: Example of a traffic scenario where the A10 north is closed. [2]

These scenarios could be seen as a kind of manual network management. They overwrite the current autonomous control systems of the actuators if that is allowed by the default behaviour of the device.

In consultation between the different organisations, each with their interest, the scenarios are built and approved. The policymakers of provinces or municipalities themselves could argue for a safer or more environmental friendly effect of traffic control, while road operators themselves or 'Rijkswaterstaat' in particular are more interested in an efficient traffic flow. This is one of the problems of traffic management. How to deal with all the wishes of all the different stakeholders? What control objective has the highest priority? Inevitably, one has to do concessions.

One of the problems in traffic management is the communication between different organisations. To align the terminology and the approach of traffic management between the different organisations a new concept named the "Landelijke Regelaanpak" (LRA) was developed [5] in 2017. The concept categorises different types of services. The services are the type of measures one can use. Within the concept of the LRA, the services are one level lower than scenarios and one level higher than just actuators. The concept divides the services into the following subcategories; decrease inflow, increase outflow and rerouting. The first 2 options are more physical options by using traffic lights in different forms. The latter option is more about informing road users for example with in-car traffic and the variable message signs above and aside roads. All services aim to improve the traffic condition for the trigger area subsequently, the surrounding areas will be negatively affected. This concept is visually illustrated in figure 1-3, 1-4 and 1-5, where one can see the services schematically.

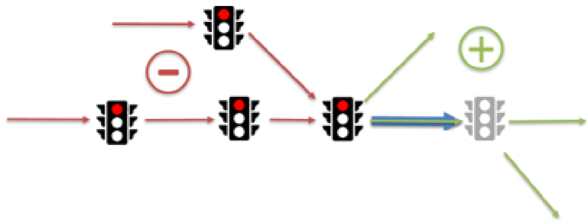


Figure 1-3: The service *decrease inflow*. Traffic lights give more red resulting in better traffic conditions at the roads after the traffic light. The traffic condition of the roads in front of the traffic light deteriorates. [Technolution]

Figure 1-4: The service *increase outflow*. Traffic lights give more green resulting in better traffic conditions in front of the traffic light. The traffic condition of the roads after the traffic light deteriorates. [Technolution]

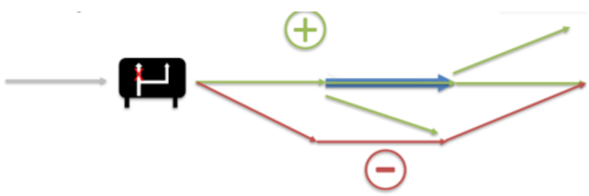


Figure 1-5: The service *rerouting*. Road users are informed about an alternative route. This results in less traffic for the original route resulting in a better traffic condition. The amount of traffic condition for the alternative road will increase resulting in a worse traffic condition over time. [Technolution]

The LRA could be seen as an automated static action plan where it is still possible to manually activate specific services by the traffic operators if necessary. The services are activated in case a certain road segment, called a link, is becoming more crowded, the traffic is influencing a control point upstream or the traffic is hitting a point of choice upstream. This way the traffic problems are propagated through the network to links with lower priority.

1-2 Relevance of network-wide traffic management

Numerous times a day traffic is operating near its maximum capacity in the Netherlands. This is a trend that will continue, see the article from the newspaper *AD* from 2016 [6] where they state: “Despite the construction of extra lanes, the Dutch are more often in traffic jams. Travel time loss has only increased since the crisis". In an article of ‘The Parool’ [7] it is Furthermore stated that "Towards 2024, traffic on the main road network will grow by 8 per cent, while total road traffic will increase by 5 per cent." Besides the possible causes for traffic problems in the future, given in the articles above, there are also other reasons for maximum capacity problems. For instance peak hours, which are foreseen reasons and will always be present. On the contrary, there are also unforeseen reasons like disruptive weather, malfunctioning or accidents which can cause capacity problems.

The given circumstances in the paragraph above underline the need for other approaches to reduce capacity problems on the road networks in the Netherlands. Network-wide traffic management could be an interesting approach. Where the goal is to find a network-wide optimum for a set of traffic control actions. This way the overall network performance could be improved. Instead of locally fixing problems and only taking into account the limits and boundaries of the local road network, a network-wide approach would try to find a global optimum set of traffic controls to propagate the traffic through the network in the best way possible.

The exchange of traffic problems between city and freeway traffic is one of the main problems when designing traffic control strategies. This exchange that happens due to the applied control strategy should be prevented [8]. Control measures taken in one of the two areas can have a significant influence on the other area [9]. This emphasises once more the possible strength of network-wide traffic control.

Another interesting observation of the current situation for managing traffic is the increasing amount of smart control applications and scenarios. Most of them are used and deployed with all satisfaction. However, a point of attention is that these applications and control options are mainly for local purposes and should still be monitored (or activated) at all times by operators. Next to that, the complexity of the problem increases by a large number of options. That is another reason why traffic centres could be helped by a network-wide approach that could support operators by advising the right scenarios or alerting on roads that need extra attention. While in the background the local automatic applications run.

1-3 Use-case

The use-case of this thesis is proposed by Technolution. The company developed a new transparent and user-friendly traffic management algorithm called the Network Manager (NM). It has been developed for the region of Amsterdam in cooperation with the city and government. The concept of the NM focuses on two aspects. First of all, it should be user friendly and easy to implement. Secondly, the control measures which are suggested or directly implemented by the algorithm should be transparent and explainable. The network manager is implemented in practice by the municipality of Amsterdam. The area of focus is the ring of Amsterdam including freeways, provincial roads and some urban roads. See a sketch of the network below in figure 1-6:

Figure 1-6: The ring of Amsterdam. This is the area of focus for the use-case of Technolution. [Own work]

In the first implementation, there were around 50 different types of measures configured in the toolbox of the network manager. Right now, it is hard for Technolution to have a clue about the actual performance and added value of the NM. They believe in their concept and strategy. However, some hard numbers on its performance lack. During this thesis, the network manager will be of great importance. It will be under reflection for the entire thesis and will be compared to a more advanced control method, Model Predictive Control (MPC). The working principle will be discussed later on.

1-4 Goals and objectives

In this chapter, the main focus of this research is shared. This is done by giving the goals and objectives. At last, the expectations of the research are shared.

1-4-1 Goals

The main goal of this research is to give Technolution a clear direction on, which approach for network-wide traffic management algorithms, they have to develop further. Which approach is in their case the most promising one and worth developing further?

To come up with a solid answer, an analysis or test framework is needed. Where the performance of multiple control methods can be tested and analysed on its performance. Therefore, a sub-goal of this thesis is: To develop a test framework that is suitable to use in future cases, where it is easy to test algorithms on their performance. It should be possible to add new algorithms and test the same algorithms for new networks and configurations.

1-4-2 Objectives

To accomplish these goals, specific and measurable objectives are needed. The objectives are formulated in the list below:

- Find a suitable simulation program.
- Build a simulation environment, with multiple network configurations.
- Find out what suitable KPIs are for analysing the performance of the control options.
- Design the architecture such that different control approaches can interact with the simulation and influence traffic control settings.
- Design an MPC controller for a network-wide control approach for small and simple networks.
- Determine the main points of attention and bottlenecks for the MPC controller in case the network becomes larger and more complex.
- Map the performance of different control methods and compare them.

1-4-3 Expectations

In this section, the expected results and outcomes of this research are shown. In the basics, the network manager is easy to configure, however, the optimisation part of the algorithm is per definition sub-optimal. It is based on prioritisation rules instead of an optimisation problem with an optimum. Where the core of MPC is about the optimisation strategy, it is expected that it could achieve better performance. As long as the optimisation problem can be (easily) solved. The different hypotheses are listed below:

- The NM is easier to configure and implement than an MPC controller.
- The NM improves traffic conditions.
- The MPC controller improves traffic conditions.
- The MPC controller exceeds the computation time of the NM.
- The MPC controller achieves better traffic conditions than the NM.

1-5 Motivation

Many reasons support the development of the NM in the first place and that support further research and validations. Next to that, there are also good reasons to investigate other methods and compare the performance. In this section, some of these motivations are shared, but also for the entire research itself. Next to that shortcomings of current solutions are given. At the last one can read the main contribution of this research.

To start with, it is really hard for Technolution to validate the working performance of the network manager in practice at the moment. Therefore simulations can prove the potential or validate the working of the algorithm positively, but also negatively. This is a good motivation for a testing framework, for now, but also for the future. Next to that, a comparison will provide extra useful information. See what certain key performance indicators do for a base scenario, a scenario where the NM is active and a scenario where an MPC controller is active. This puts the performance in perspective, see for instance [10].

One of the first reasons to develop the NM for Technolution was driven by the ease of implementation. If the algorithm should be sold to different municipalities or governments it should be rather easy to configure the algorithm for new cases.

Furthermore, the algorithm should be real-time active so computational complexity is a topic to deal with. If computational times get out of hand it is not realistic anymore for real-time implementation.

1-5-1 Shortcomings of current solutions

Transparency
One of the main problems with the smart control applications that are not implemented yet is often the lack of transparency. While this is a really important aspect of the public sector. Since choices that have been made have to be explainable. Road operators are held accountable by their supervisors whom themselves are driven by political policies. In case of emergencies or when things go wrong road operators and traffic centres are held responsible. That is one of the reasons why there is a certain restraint to techniques that are not transparent. Black-box or grey-box methods do not have the preference.

Deduction
Because of the reason above, road operators want to understand what the outcome is of a smart control algorithm. Mainly, because they are still held responsible. For that reason, the outcome of the network-wide control method must be also easy to deduce and understand. Current results of smart control options are not easy to deduce yet.

Implementation and configuration
Most of the other smart control approaches or algorithms are harder to configure for new networks. Technolution aims at easy implementation with the NM.

1-5-2 Main contribution

The main contribution of this research will be a framework analysing different control methods in one framework or (code) environment. Furthermore, this is done from a practical viewpoint where the use-case has practical and commercial applications/implementations this use-case will be explained in the next chapter. For this reason, options and results are discussed in a way where we consider the user-friendliness of implementation next to the performance on traffic KPIs and the support a certain approach could give to daily traffic operators.

For companies active in designing smart traffic control algorithms it is hard to test the real performance in practice. For single intersections, this is somehow doable and measurable. However, as networks enlarge this becomes harder and harder. Therefore, companies would be helped by testing frameworks, where they can test new algorithms on different situations and scenarios. See for instance paper [10], this shows there is indeed a desire for this kind of framework, which can help companies in making the right choices.

Furthermore, a realistic study on the feasibility of an MPC controller for a network-wide traffic control problem is done. Which takes into account ease of implementation, computation time and performance.

1-6 Research questions

In this section, the goals and objectives are summarised and formulated into the main research question and sub research questions.

Main research question

- Can a user-friendly, transparent, network-wide traffic control algorithm achieve a promising performance compared to more advanced although nontransparent control methods for large mixed city and freeway networks?

Sub-research questions

- How to build an evaluation framework for comparing different network-wide traffic control methods?
- What are useful KPIs for such a framework?
- How to design an MPC controller for a network-wide control approach for small and simple networks?
- How suitable is the MPC controller approach in case the network becomes more complex?

Chapter 2

Background information

Chapter two informs the reader about the main lessons from the literature study done in advance. It provides the reader with the necessary background information. This is however done briefly, for more in-depth information the reader is referred to the scientific references and the literature study itself [11]. That study researched performance measures, traffic modelling, control approaches and simulating within the field of traffic. In that study, the following questions are studied and answered:

- Which key performance indicators are useful for evaluating (complex large-scale) networks that include city and freeway traffic?
- How to model (complex large-scale) networks including city and freeway traffic?
- How to control (complex large-scale) city and freeway traffic systems?
- How to simulate (complex large-scale) networks including city and freeway traffic?

In the context of this research 'large' implies at least the size of a small city. A large-scale network of between 50 and 100 links which could be as long as 100 metres or as long as 2 kilometres. Each link could consist of multiple sub-segments if there are different data points. Complex implies different types of services and an amount of at least 20 or 30. Part of this research is focused on a use-case for the city of Amsterdam. The network is reduced to around 100 links which consist together of approximately 2500 sub-segments and 50 services.

Below are the main findings in short of the done literature study.

2-1 Key performance indicators

To analyse and calculate the performance, one needs road traffic data. Especially in the case of real-life situations, it is crucial which information is available to do an analysis. In [12] different collection methods are discussed. The purpose is to discover which key performance indicators are suitable for evaluating (complex large-scale) networks that include city and freeway traffic. Taking into account the usability for real-life situations and not just in the simulation environment.

In the case of Technolution, available data collection methods are Floating Car Data (FCD) and induction loops. Given the available data and the possibilities of KPIs, there are a few suitable candidates. Promising candidates are vehicle lost hours, Total Time Spent (TTS), the travel time or average speed for the main route and the Level of Service (LoS). Vehicle-lost-hours is a widely used parameter for network-wide performance, it remains however a big generalisation and possible improvements could be diminished by spreading it out over the entire network. A hundred vehicle lost hours could mean 100 vehicles losing 1 hour or 1 vehicle losing 100 hours. The total time spent is also a network-wide KPI, this parameter tells the total amount of travel time all the vehicles together spent in the network. Travel time or average speed for a certain route is a more focused KPI on important or objective routes for control. The LoS is working on a delay component calculated by dividing the vehicle lost hours by the production of a segment, see equation 2-1.

$$\mathrm{D} = \frac{\mathrm{VLH}}{\mathrm{P}} \tag{2-1}$$

If there is no data or information on the intensity of a segment, the level of service is calculated by dividing the travel time - travel time with free flow speed by the length, see equation 2-2.

$$\mathrm{D}^* = \sum_{1} \frac{\max(T_i - T_{i,free}, 0)}{L_i} \tag{2-2}$$

Subsequently the delay factors of the segments are added by weight to get a total delay factor for a link, see equation 2-3.

$$\mathrm{D}_T = \frac{l_f}{l_f + l_s} D_f + \frac{l_s}{l_f + l_s} D_s \tag{2-3}$$

Where D_T is the total delay of a link consisting of the delay of all the segments in that link. l_f and l_s are the lengths of the segments with respectively flow and speed data and only speed data. D_f is the delay of the segments calculated by equation 2-1 and D_s is the delay of the segments calculated by equation 2-2. Lastly, this value is translated into one of the LoS A to F.

In the case of simulation, there is more variety of data and information available for extra KPIs. However for the simplicity and usability for Technolution afterwards, the first focus is on the KPIs stated above. Besides those traffic orientated performance indicators, the control strategies can also be compared by computation time, ease of implementation and transparency. Although these are more subjective and non-measurable indicators. At last, there are also numbers available on accidents and safety, data can be found on [13] [14] [15], these can be used as well for evaluation of performance.

2-2 Traffic modelling

Regarding traffic modeling, different model types have been researched in the literature survey. Like, a macroscopic second-order model METANET [16], a first-order Cell Transmission Model (CTM) [17], a Store and Forward Model (SFM) [18], an extension like a BLX-model [9] [19], or the S-model [20]. Furthermore, different categories of models are given and different types of models based on level of detail [21] [22], see for example table 2-1 below on level of detail.

Table 2-1: Overview of different model classifications based on level of detail.

Classification	Level of detail	Variables	Computation time	Complexity
Microscopic	+	Individual	-	-
Macroscopic	-	Aggregate	+	+
Mesoscopic	+/-	Both	+/-	+/-

The aim was to find a suitable model for traffic flow for (complex large-scale) networks including city and freeway traffic. During the literature study, rather quick was concluded that METANET [16] [23] is a suitable candidate for the use case of Technolution. It is a macroscopic traffic flow model and is frequently used within the field of traffic control. It serves multiple configurations, control strategies and control approaches. Furthermore, the trade-off between speed and accuracy is good [24]. For the complete model description the reader is referred to [23] and [16] or the literature survey [11].

2-3 Simulations

To answer the question on, how to simulate (complex large-scale) networks including city and freeway traffic, multiple simulation categories and programs have been compared and examined [25] [26] [27]. The different software programs can be found in table 2-2 below.

Table 2-2: Properties of commonly used simulation programs.

Program	Type	Scale	Availability	Visualisation
OmniTRANS [28]	Four-step	macroscopic	Commercially	2D
VISUM	Four-step	Macroscopic	Commercially	2D
VISSIM	Traffic flow	Microscopic	Commercially	3D
AIMSUN	Traffic flow	Micro, Meso & Macro	Commercially	3D
SUMO [29]	Traffic flow	Microscopic	Open-source	2D
MITSIM Lab	Traffic flow	Microscopic	Open-source	2D
MATSim	Assignment model	Microscopic	Open-source	2D
OpenTrafficSim	Traffic flow	Micro, Meso & Macro	Open-source	2D

Requirements for this thesis and wishes of Technolution are, dynamic control, dynamic model, open-source, compatibility with different programming languages and commonly used in the scientific field. Taking these requirements into account SUMO turned out to be the best option. SUMO is a microscopic model that simulates the behaviour of changing lanes and following cars separately. Where the speed of individual vehicles is determined based on the vehicle immediately in front of it and changing lanes is determined by valid paths and the distance upfront which allows for tactical changes. For more detailed information see [30] and [11]. SUMO has a broad community and a decent track record. Furthermore, is it well-known in the academic field of traffic [31]. This makes SUMO an established open-source software program.

2-4 Traffic control

The goal was to find answers on how to control (complex large-scale) city and freeway traffic systems. Mainly used actuators in the field of traffic control are traffic lights, VMS, VSL and in-car advice. In the literature survey multiple concepts of control in the field of traffic have been researched, see table 2-3.

Table 2-3: Overview of discussed control methods, papers in which those methods are discussed and their key features.

Category	Papers	Short explanation
Feedback	Pasquale et al. [32]	Measurements of the system are used to calculate the control measure to reach the reference output.
Feed-forward	Frejo et al. [33]	Feedforward is typically used together with feedback in case feedback alone cannot reject continuous present disturbances.
Model Predictive Control (MPC)	Hegyi et al. [3], Lin et al. [34], Frejo et al. [35], Ye et al. [36]	Future control measure are calculated by optimising an objective function based on measurements and model predictions.
Game Theory	Groot et al. [37]	Is about modeling strategic decisions and interactions between two or more players, where you can have leaders and followers for example.
Reinforcement Learning	Walraven et al. [38]	Tries to find the optimal policy by learning from rewards.
Event-triggered control	Pasquale et al. [39] [40]	Able to reduce unnecessary calculations in comparison to time-triggered control by making control actions triggered by events.

One of the most used control strategies for network-wide traffic control, where the network is large, is MPC. The ability of MPC to deal with constraints and disturbances and to find an optimum control input is useful. Multiple papers have proven it to be a suitable approach for the use case of Technolution for this thesis research. For a detailed description on MPC the reader is referred to [11] [41]. In short, MPC calculates optimal control inputs by predicting the output based on model calculations for a certain horizon. To do so it solves each step an optimisation problem.

Furthermore, the network manager is discussed in-depth, for more information the reader is referred to [11]. The basic principle of the network manager is based on the level of service of certain road links. The LoS can be improved by measures, which have positive and negative areas of influence, which improve or decrease the LoS. The core of the NM algorithm can be seen as a rule-based greedy algorithm, that determines based on prioritisation rules which measures are activated.

Chapter 3

Design and methodology

In the previous two chapters, the goal and research question has been shared and the reader is acquainted with the necessary background information. To answer the research question and to fulfil the goal, different environments and aspects have to be brought together. This chapter informs the reader on the design and methodology of the research. In the first section, the test framework is given. In the second section, the networks and set-ups are discussed. Section three focuses on the demand and routes. In section four the model and service configuration are given, for both the controllers. At last, the problem formulation is given for the optimisation strategy.

3-1 Test framework

For simulating the traffic, SUMO is needed. For controlling the traffic, the NM or the MPC controller is used. This paragraph explains the framework that brings all the aspects together.

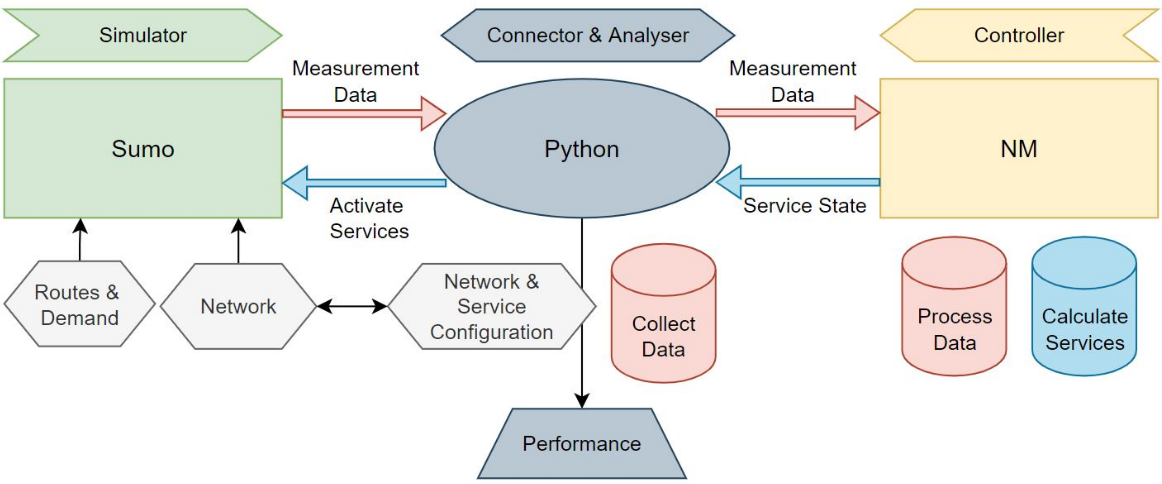


Figure 3-1: NM test framework, including all aspects of the testing process. [Own work]

As long as the controller can communicate with Python a (new) algorithm can be incorporated into the framework suggested and thus be tested on performance. See figure 3-1 for the overview with the NM controller in yellow. With the help of TraCI Python can communicate with SUMO. TraCI can retrieve data and it can change settings. In advance of the simulation, the networks have to be built with the SUMO net-editor and one has to define the route & demand files. Furthermore, in the case of the NM, the networks have to be configured in a Python file and the services have to be configured, such that these can be communicated in the right format to the NM controller. The algorithm is written in Kotlin and developed by Technolution.

When the simulation runs, the current traffic data is retrieved every 60 seconds by the Python agent. In the case of the network manager, the data are FCD and densities, calculated with the help of induction loops. If the agent has collected the data it is sent forward to the controller in the right format. The NM agent runs its algorithm and calculates the new service states. These are again collected by the Python agent and subsequently, it (de)activates the services in SUMO and the simulation continues again for 60 seconds and so on. After the simulation, the key performance indicators are collected and calculated. Furthermore, useful plots can be made.

See figure 3-2 for the overview with, in orange, the MPC controller. The outline is almost the same, with some small differences.

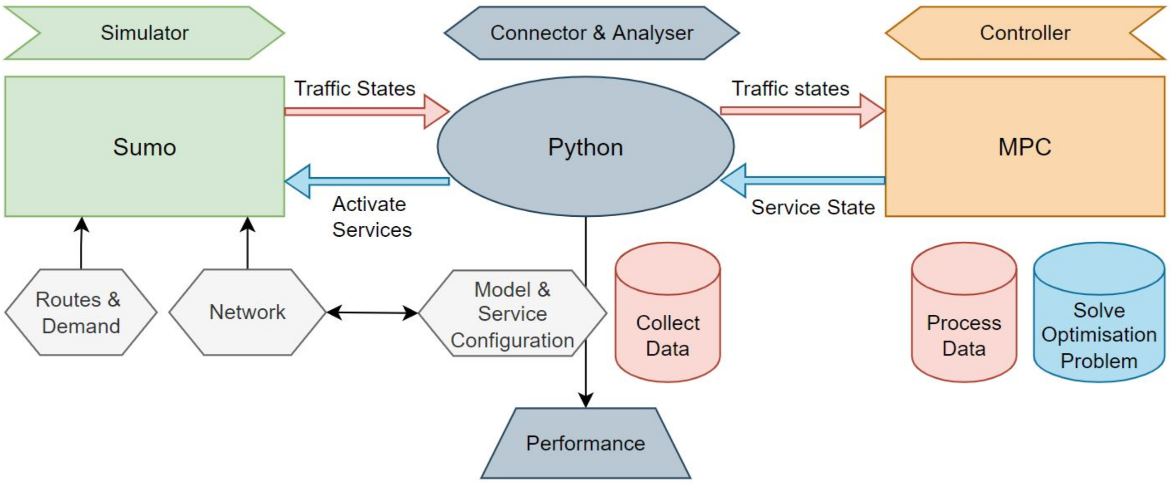


Figure 3-2: MPC test framework,including all aspects of the testing process. [Own work]

In this case, the Python agent communicates the traffic states to the controller, which is based on the same type of measurements, however in a different mathematical format. Next to that, the MPC controller solves an optimisation problem. This problem will be explained in further detail later on in section 3-5. Last but not the least, one has to configure the METANET model to the specific network. This is probably the most time consuming and important part. Determining the right model equations and connections including the model parameters. In most cases, one can take standard model parameters. However, to improve model quality one could try to find the network or case-specific parameters. This will also be explained further in section 3-5.

Test possibilities

Once the framework is built, it is easy to see the possibilities there are, this paragraph will discuss some of them. For example, there is the possibility to test algorithms for different networks. One can add a new network by drawing the connections in the net-editor. Next, one has to redefine the specific routes and demands for this network and do the configurations in Python for the network and services.

Furthermore, within one network the performance for different traffic demand scenarios can be tested as well. This requires less work but provides useful information. Within the framework, it is easy to switch between demand files and run different simulations.

Besides that, one can also change service configurations. What is the actual change on the street when activating a service? Or in the case of the NM, what happens if we change negative or positive/trigger areas for instance?

On the controller side, it is possible to test different algorithms by changing the controller block. If the controller already exists it needs to be able to communicate with Python to be incorporated into the network. If it still needs to be designed it is advised to design the controller in a Python environment.

3-2 SUMO networks & set-up

One of the most important aspects of realistic simulations is having a realistic representation of the real case. To achieve such a representation the simulation needs to be calibrated with data like OD matrices, road capacities and commonly used routes. However, in this case, the information lacks and the focus is different. Besides, if only a scientific proof-of-concept is needed, one could come up with a network that contains the most essential elements it wants to take into account. For example in paper [9] where they choose a network that was small enough to analyse and interpret the results on intuition. While it was still possible to test the most relevant effects. The same approach will be chosen in this research, capturing the most important aspects in as small and as simple as possible networks. In the following paragraphs, the different networks are introduced and explained what the added value of this network could be.

3-2-1 Network 1 - simple freeway

This is the most basic network, called the simple freeway network, see figure 3-3 for an illustration. It is a straightforward network with double lane roads. There is only one direction of 8 kilometers plus the on-ramps. The allowed traffic speed is 100 km/h.

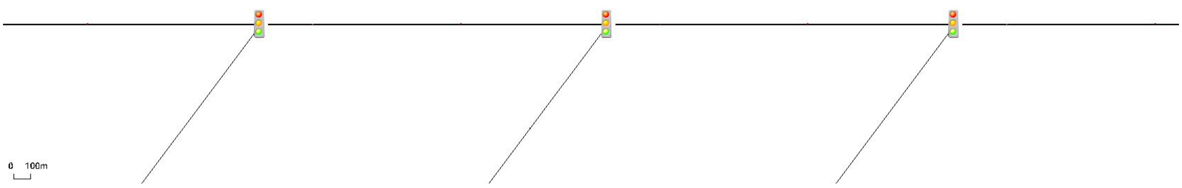


Figure 3-3: Simple freeway network in SUMO. [Own work]

All the network parameters are summarised and can be found in the tabular 3-1 below.

Table 3-1: Simple freeway network parameters.

Parameter	Value	Unit
Speed	100	[km/h]
Direction	single	[-]
Lanes	1-3	[-]
Traffic	freeway	[-]
Total length	14	[km]

There are on-ramps at each 2 km, with an additional insertion lane for 300 m, see figure 3-4 for an impression. The on-ramps are provided with ramp meters available for control. The effect can be determined by changing the green en red times for the on-ramp lane. The exact configuration of this control will be discussed in section 3-5.

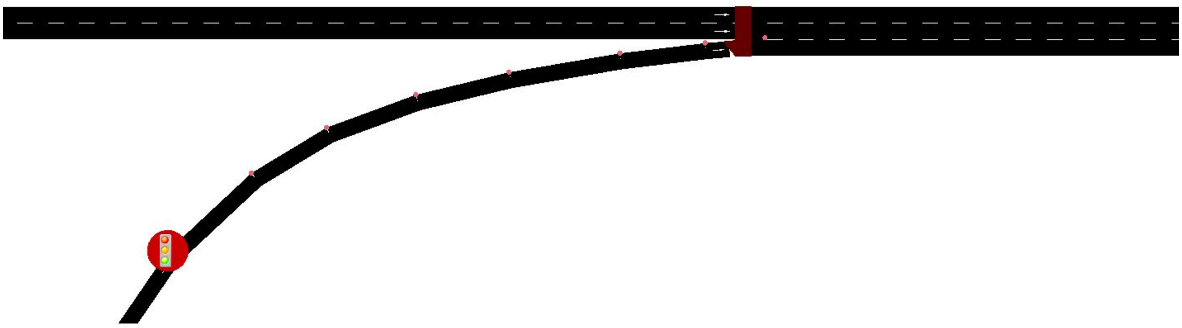


Figure 3-4: Simple freeway network close-up. [Own work]

Furthermore, there is a different intended goal for each network. For network number 1 the goal is to test the basic principles of the network manager and check whether the framework works as expected. Besides that, this network is suitable for a rather simple implementation for MPC but still allows for a good qualitative analysis between the two.

3-2-2 Network 2 - simple freeway plus

This network is based on the first network, for that reason, it is called the simple freeway plus network, see figure 3-5 for an illustration. It is a basic network with 2 different routes for the same origin-destination pair. Route 1 is straightforward and 10 kilometres long and route 2 is a detour of 11 kilometres long. Furthermore, there is 1 on-ramp that heads for the same destination as the other traffic. The speed limit is 100 km/h.

Figure 3-5: Simple freeway plus network. [Own work]

The parameters of the simple freeway plus network are summarised in the tabular 3-2 and can be found below. These details can be found in the close-ups in 3-6.

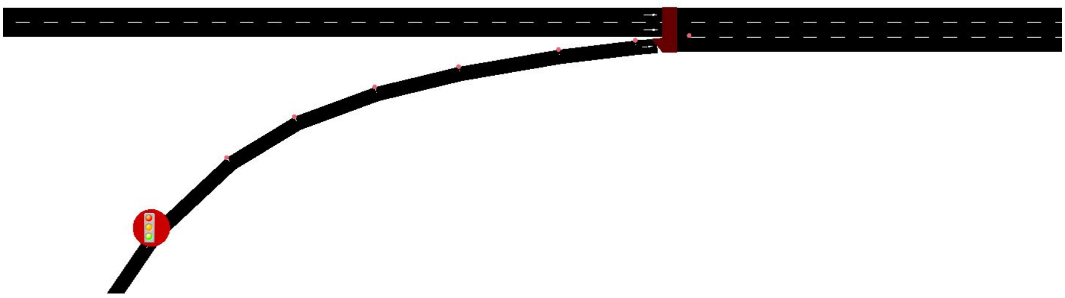
Table 3-2: Simple freeway network parameters.

Parameter	Value	Unit
Speed	100	[km/h]
Direction	single	[-]
Lanes	2-4	[-]
Traffic	freeway	[-]
Total length	21	[km]

The beginning and the end of the network consist of 4 lanes, the rest of the network has 2 lanes, except for the link where there is an extra insertion lane for the on-ramp. The on-ramp itself has a single lane.

(a) Close-up separate roads

(b) Close-up unite roads



(c) On-ramp

Figure 3-6: Simple freeway plus network close ups. [Own work]

For network 2 the goal is to test the principle of the NM of using services that overlap in the influence area. Next that this network also shows how the MPC performs with different types of services.

3-3 Demand & routes

In this section, the setup of the sumo route and demand files will be explained. The format of these files is XML. First, some definitions, for SUMO a trip is a vehicle movement from one place to another defined by the starting edge (street), the destination edge, and the departure time. A route is an expanded trip, which means, that a route definition contains not only the first and the last edge, but all edges the vehicle will pass.

3-3-1 Routes

Determining routes can be done in several ways with SUMO. The most basic approach is to determine the route files completely by hand for each vehicle separately, however for larger networks and multiple cars this is not an option. The next possibility is to give trips of flows, where one could leave out the specific route and just define a starting edge and ending edge. Where in the case of trips one still has to explicitly name each car. However, for flows, one can combine multiple cars, as long as they make the same trip. The cars are uniformly spread over a certain time interval where the frequency is determined by the flow. Now the routes within the simulation can for example be determined by *duarouter*, which can determine vehicle routes using shortest path computation. *Duarouter* can also be used to calculate a dynamic user equilibrium when called iteratively. Other options are also available, like using turning ratios and the module *jtrrouter* or using OD-matrices and the module *od2trips*.

However, when the network is simple and there is only one route available or one specifies all flows for the possible different routes, these flows can be fed directly to SUMO. This is the case for network 1, there are no other routes available for a certain origin-destination pair. For network 2 this is done by specifying the flow for each different route option. The advantage of this is that the simulation will not be smart and the different scenarios can be tested scientifically and comparably. The trip files that are computed, are completely identical when used for the different control scenarios; no-control, full-control, NM-on and MPC-on. Later on in the process, the input will be slightly changed so that the trip files will differ between different simulations, but not between the different control options. Such that a more sophisticated and weighted average performance can be computed and calculated.

3-3-2 Demand

The different scenarios are based on the critical density ρ_c, of the network for certain roads. See figure 3-7 for a visual interpretation of this point. Before this point traffic drives fluently and beyond this point traffic speed decreases and traffic jams start to occur.

The demand is given by specifying the flow, and the number of vehicles per hour. Common numbers in practice and real-life situations are approximately 2000 vehicles per hour per lane. When the number of lanes increases the possible flow does not increase with the same number. For this theoretical simulation case, the number is expected to be a little higher, since it is an ideal lab situation. There are possibilities to change a lot of different vehicle and driver settings to match certain cases. However, the possibilities are enormous and for this research, this aspect is not of interest. Next to that, there is also no data available that could be matched. For this research, it is more important the different cases and simulations are somehow realistic and are comparable to each other. For that reason rules of thumbs are used together with a heuristic approach to find the critical density for where the network manager and the MPC controller can still make a difference.

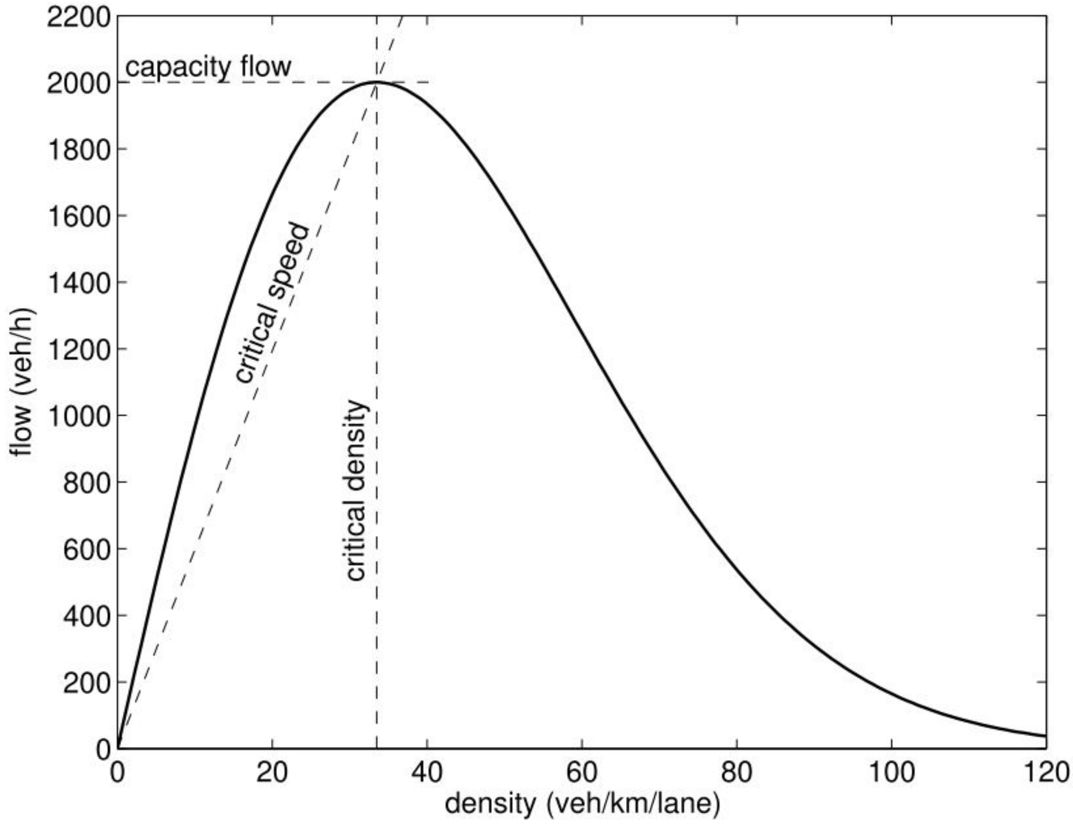


Figure 3-7: Flow-density relation. [3]

Scenarios

For the first scenario, the demand and flow numbers are chosen such that, the density on the roads is below the critical density. The amount is picked by decreasing the flow that leads to the critical density by approximately 10%. The demand and flow for the second scenario will be determined with a heuristic approach by observing and analysing different amounts. Where the traffic is decreasing speed and close to cause traffic jams. For the third and last scenario, the demand and flow are increased by approximately 10%. See the actual demands below in tabular 3-3 and 3-4 for respectively network 1 and 2.

Table 3-3: Demand flows in veh/h for three scenarios for network 1.

	Demand	Value
Scenario 1	main route	2000
	ramp 1,2,3	800
Scenario 2	main route	2300
	ramp 1,2,3	800
Scenario 3	main route	2600
	ramp 1,2,3	800

Table 3-4: Demand flows in veh/h for three scenarios for network 2.

	Demand	Value
Scenario 1	route 1	3600
	route 2	4000
	ramp	800
Scenario 2	route 1	4000
	route 2	4000
	ramp 1	800
Scenario 3	route 1	4400
	route 2	4000
	ramp	800

Iterations

To give the results some extra validity the averages are taken over 5 iterations. Per control case. To get this randomisation one can add '–random' to the SUMO command line. The simulation uses random number generators to decouple different simulation aspects to achieve randomness when loading the vehicles into the network. This way it will generate for example different speed factors and distributions.

The variance over these iterations can be calculated with formula 3-1 and the standard deviation is the square root of the variance.

$$\sigma^2 = \frac{\sum_{i=1}^{n} (x_i - \mu)^2}{n} \tag{3-1}$$

Expectations

The expected performance results for each of the scenarios and the different control methods, is showed in table 3-5 below.

Table 3-5: Expected performance of the different scenarios and control cases.

	intensity	No-control	NM-on	MPC-on
Scenario 1	$-10\%\ \rho_c$	Good	Good	Good
Scenario 2	ρ_c	Bad	Good	Good
Scenario 3	$+10\%\ \rho_c$	Worse	Bad	Bad

3-4 Model & service configuration

This section discusses the configuration of the different models and their accompanied services. First, the network manager case will be discussed and next the MPC approach.

3-4-1 Network manager

Working principle

The Network Manager (NM) is focused on network-wide traffic management. One of the main focus points of the NM was to develop a network-wide control algorithm that remains understandable and user-friendly for the road traffic operators. The model of the NM is a simple high-level one. The model is based on the Level of Service (LoS) of links. These links are split into segments from which separate data can be obtained. This data is obtained by induction loops and FCD. The model is about changing this LoS with control measures such as traffic control lights (more green/red time), ramp metering and rerouting by variable message signs or in-car advice. These control measures are called services, which have positive and negative effects on specific links. The model divides the services, into the following subcategories; decrease inflow, increase outflow and rerouting. For a visual impression see figures 1-3, 1-4 and 1-5. However, in the model, there is no time dependency. So the real impact of these measures is not calculated and cannot be predicted. For each new time interval of 60 seconds, the LoS is calculated. In case the LoS is under a certain threshold, services are (again) triggered and want to be activated. Next, the NM will run the greedy algorithm to determine which services will be activated. This is done based on prioritising the type of service, LoS and the type of interaction in overlap services have. The services will be activated and after 60 seconds the new data will be retrieved to calculate the current LoS.

Implementation network 1

With the ideas of the NM in mind the network is divided into 10 links and some links have 2 segments. The services are 3 ramp meters, inflow reduction, located at the end of the on-ramps. The trigger links (blue), positive links (green) and negative links (red) configuration can be seen in figure 3-8 below.

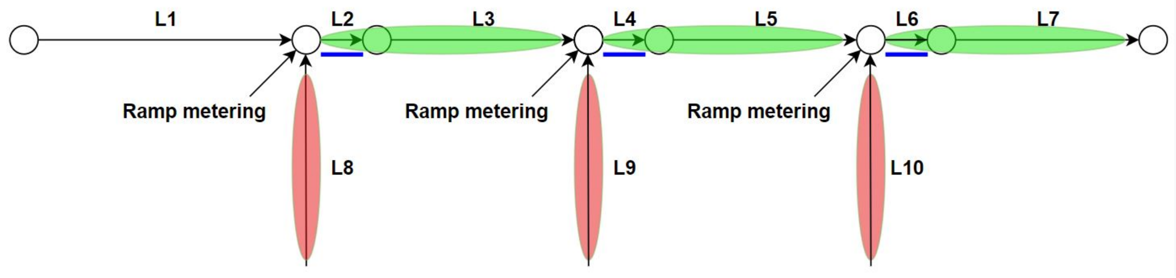


Figure 3-8: The NM services configuration of the simple freeway network. [Own work]

Implementation network 2

This network is divided into 7 links consisting of segments within a range of 1 to 6. There is 1 on-ramp with ramp metering control, S001. Next to that, there is a rerouting service, S002. See figure 3-9 below for a visual configuration of the network including the configuration of the services. The trigger link is blue, the positive links are green and the negative links are red.

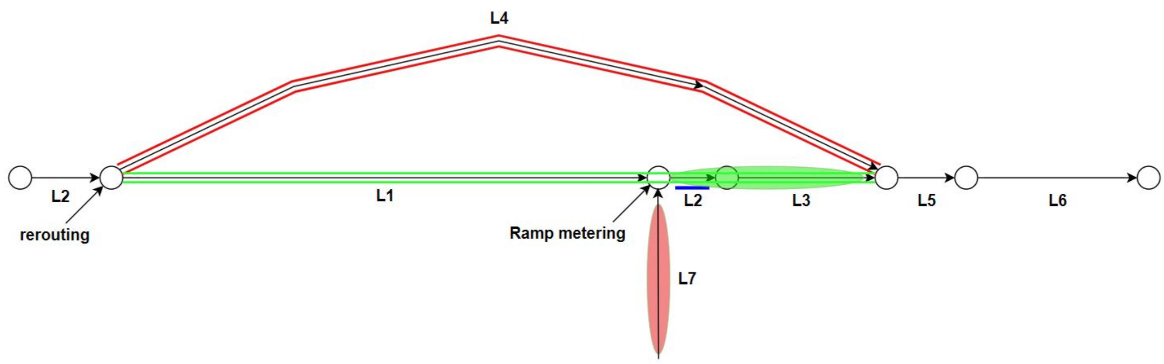


Figure 3-9: The NM service configuration of the simple freeway plus network. [Own work]

3-4-2 Model predictive control

This section discusses the configuration of the METANET model used for MPC for the different networks. But first, the working principle of MPC is given briefly. Furthermore, the different control actions, also called services in NM terminology, are given for the different networks.

Working principle

Model predictive control is based on a system model which it tries to use to its benefit in order to calculate the optimal control input for the next step, see figure 3-10.

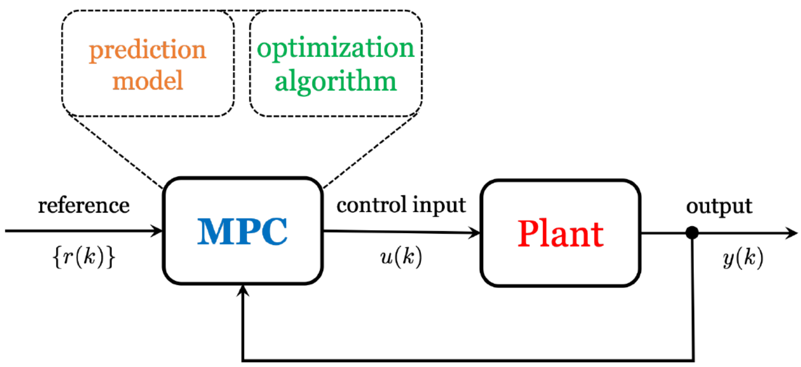


Figure 3-10: Block diagram of a control scheme with a MPC controller. [4]

It does this with the help of the receding horizon principle. This means that k control steps ahead are calculated, while only the first control step is implemented, see figure 3-11:

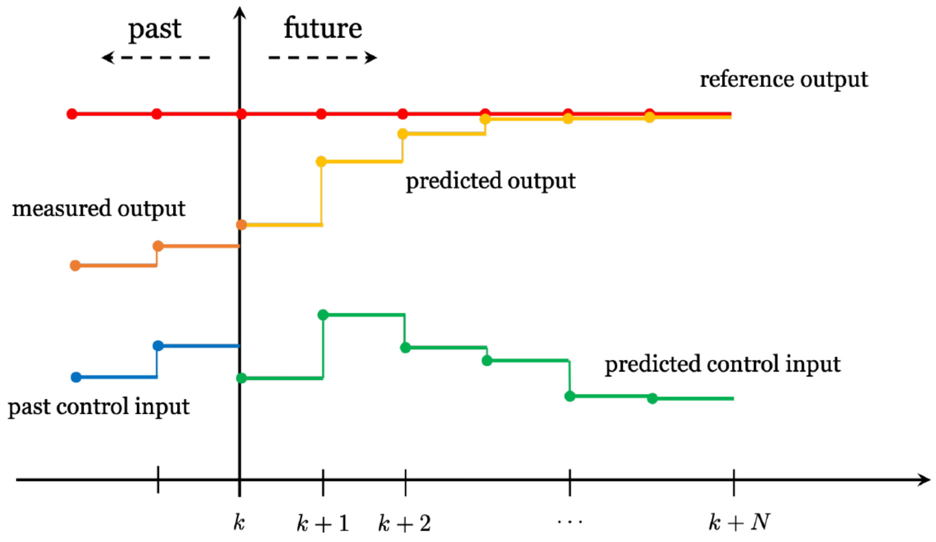


Figure 3-11: Illustration of the receding horizon principle. [4]

This process is repeated each time step. This makes it a robust method that can deal with uncertainties and disturbances. To calculate the optimal control input an optimisation problem is solved. This will be explained in section 3-5.

For further information on the principles of MPC see chapter 2, paper [11] or [41].

METANET configuration

As was already discussed in chapter 2 the model used for the MPC approach will be the macroscopic model METANET. It is suitable just for freeway networks. One has to make use of other models and combine those to make it work for mixed traffic. METANET describes a network as a directed graph, where each freeway link has uniform characteristics, e.g. the number of lanes, no big changes in geometry or no off- or on-ramp. Each link has separate model state variables, which are density ρ and speed v. The model variable for the waiting queue of each on-ramp is w. See tabular 3-6 below for an overview.

Table 3-6: The state variables.

Variable	Name	Unit
ρ	density	[veh/km/lane]
v	speed	[km/h]
w	waiting queue	[#]

How these variables evolve over time is given by the following three equations 3-2, 3-3 and 3-4:

$$\rho_i(k+1) = \rho_i(k) + \frac{T}{L_i \lambda_i} \left(q_{i-1}(k) - q_i(k) \right) \tag{3-2}$$

$$v_i(k+1) = v_i(k) + \frac{T}{\tau}\left\{V(k)] - v_i(k)\right\} + \frac{T}{L_i}v_i(k)\left[v_{i-1}(k) - v_i(k)\right] + \dots$$
$$\dots - \frac{\mu \cdot T}{\tau \cdot L_i}\frac{\rho_{i+1}(k) - \rho_i(k)}{\rho_i(k) + \kappa} \tag{3-3}$$

$$w_r(k+1) = w_r(k) + T\left(D_r(k) - q_r(k)\right) \tag{3-4}$$

Where, τ, κ and μ are all model parameters, T is the time step, L_i is the length and λ_i is the number of lanes. The main principle of METANET is based on the law of conservation of vehicles. Meaning that the density of a segment at $k+1$ equals the density at k, plus the inflow of the upstream segment and minus the outflow of the segment itself, see equation 3-2. These flows of certain links are calculated in the way showed by equation 3-5.

$$q_i(k) = \rho_i(k) \cdot v_i(k) \cdot \lambda_i \tag{3-5}$$

The speed v for $k+1$ is equal to the speed at time step k plus a relaxation term, a convection term and a anticipation term, see 3-3. For the relaxation term, the desired speed needs to be calculated given by equation 3-6. For the convection term, the upstream speeds are needed and for the anticipation term, the downstream densities are needed.

$$V\left(\rho_i(k)\right) = v_{\text{free}} \cdot \exp\left[-\frac{1}{\alpha}\left(\frac{\rho_i(k)}{\rho_{\text{crit}}}\right)^{\alpha}\right] \tag{3-6}$$

Where α is a model parameter, ρ_{crit} is the critical density at which the traffic flow is maximal, see figure 3-7 and v_{free} is a network (or link) specific parameter for the free-flow speed.

The state variable w_r for the number of cars in the waiting queue at $k+1$, is calculated by equation 3-4. Where $D_r(k)$ is the demand of the on-ramp and $q_r(k)$ is the flow of the on-ramp, which is calculated by equation 3-7:

$$q_r(k) = \min\left[u(k)Cr,\ Dr(k) + \frac{\omega_r(k)}{T},\ Cr\frac{\rho_{\max} - \rho_i(k)}{\rho_{\max} - \rho_{\text{crit}}}\right] \tag{3-7}$$

It takes a minimum of 3 quantities. Firstly, the maximum flow allowed by control. Secondly, the available traffic in time-period k, equal to the waiting queue plus the demand of the on-ramp. Thirdly, the maximal flow that could enter the freeway, because of the mainstream conditions. $C_r(k)$ is the capacity of the on-ramp, $\rho_{\max}$ is the maximum density, and $u(k)$ is the control rate, a value between zero and 1.

In case there is an on-ramp connected to a link one adds the flow of the on-ramp given by equation 3-7 to the original equation 3-2 for calculating the $\rho(k+1)$ resulting in equation 3-8:

$$\rho_{m,i}(k+1) = \rho_{m,i}(k) + \frac{T}{L_m\lambda_m}\left(q_{m,i-1}(k) - q_{m,i}(k) + q_r(k)\right) \tag{3-8}$$

Implementation

Most of the variables given in table 3-6 can be directly retrieved from SUMO. This is true for speed per segment and for the number of cars, which we can divide by the length of a segment to get the density. For the waiting queue, we only need halting vehicles. Unfortunately, SUMO uses a low speed limit of 0.1 [km/h]. For this implementation cars that drive slower than 5 [km/h] are added to the waiting queue and considered halting vehicles.

See 3-12 and 3-13 for a directed graph representation of network 1 and 2.

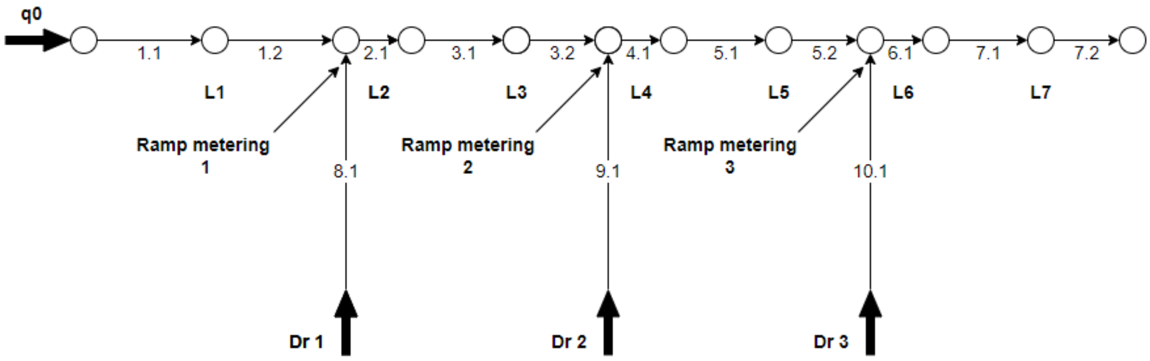


Figure 3-12: Simple freeway network as a directed graph. [own work]

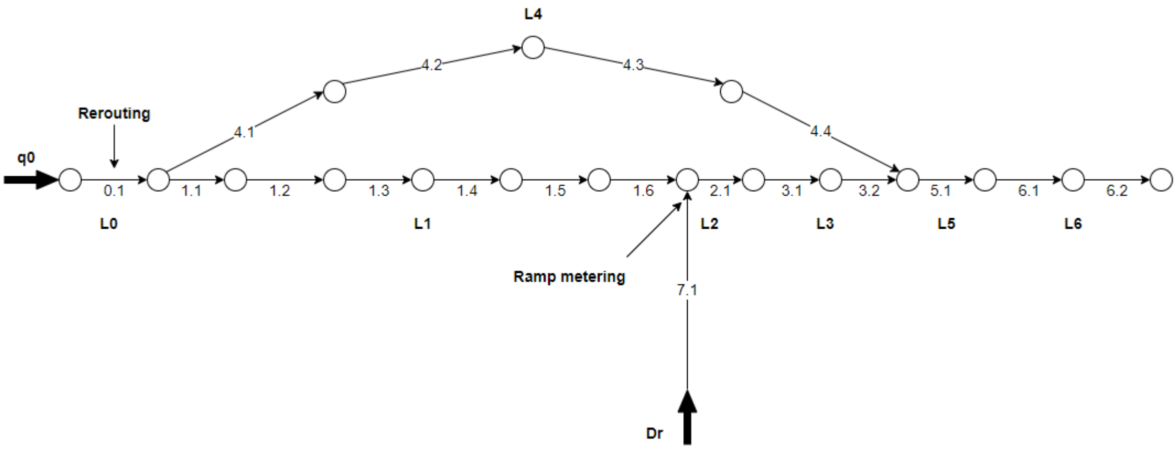


Figure 3-13: Simple freeway plus network as a directed graph. [own work]

The free flow speed of each network is determined via a SUMO simulation with very little demand. Such that the speed a car drives is only dependent on its own desire and geographical reasons. See the tabular 3-7 below:

Table 3-7: Free flow speed in [km/h].

Network	Variable	Value
1	v_{free}	95.8
2	v_{free}	97.23

The model parameters of METANET are set as they are well known in the scientific field and literature [42] [3], see tabular 3-8

Table 3-8: Model parameters.

Variable	Value	Unit
T	10	$[s]$
τ	18	$[s]$
κ	40	$[veh/km]$
μ	60	$[km^2/h]$
ρ_{max}	180	$[veh/km]$
ρ_{cr}	33.5	$[veh/km]$
α	1.87	[-]

T is the time step used for the simulation of the traffic flow, which is typically 10 seconds. The minimal link length for METANET is thereby 278 meters when calculating with a speed of 100 kilometers per hour. The links with the insertion lane and on-ramp connection are 300 meters, this meets the restriction a car cannot skip a link within a time step. Smaller time steps are not considered due to computation time and with larger time steps the performance deteriorates.

Tuning horizons

With the time step set, the horizons for model predictive control can be chosen, where N_p and N_c are respectively the prediction horizon and the control horizon. Unfortunately, the conventional rules for selected suitable horizons cannot be applied. We would need longer horizons to capture the traffic behaviour and specifically to observe and predict the influence of the control measure taken. With heuristic reasoning and testing the right horizons have been chosen for each network. The prediction horizon N_p needs to be as long as the typical travel time is from point of control to the end of the network. Otherwise, the effect of the vehicles that were influenced by measures is not taken into account during the optimisation. Due to computational complexity, N_p is tried to keep as small as possible. For network 1 a typical travel time of 309 seconds is calculated based on a travel distance of 6km and a speed of 70 km/h. For network 2 a typical travel time of 565 seconds is calculated based on a travel distance of 11km and a speed of 70 km/h. These values are taken as a starting point and after some further heuristic testing the final control horizons are set and can be found in the tabular 3-9 below:

Table 3-9: MPC horizon settings for network 1 and 2.

Network	Variable	Value
1	N_p	35
2	N_p	60
1	N_c	5
2	N_c	6

Regarding the MPC control horizon N_c, heuristic tests started with the default and a minimum value of 2. Next, the value is increased step by step. The longer the control horizon the more complex the optimisation problem will become and therefore a longer computation time is required. Within a good trade-off between performance and computation time, the N_c is set to 3 for the simple freeway network and 7 for the simple freeway plus network.

3-5 Optimisation problem MPC

In this section, the optimisation problem for the MPC problem will be formulated. Next to that, the strategy on how to solve this problem will be discussed in a subsection.

In the literature study multiple key performance indices have been studied. One of the most used KPIs is the Total Time Spent (TTS), also commonly used as objective function for a MPC strategy. The short variant of the optimisation problem is given below in equation 3-9.

$$\begin{aligned} \min_{\mathbf{u}(k)} \quad & \text{TTS} \\ \text{s.t.} \quad & \text{system dynamics} \\ & \text{system constraints} \end{aligned} \tag{3-9}$$

It is important to realise that this metric is a network-wide metric, therefore certain cars or segments of the network will be sacrificed for the sake of the overall performance. This is also the challenge of the problem.

The goal is to minimise the TTS in the network by all the cars over the total time horizon, subject to the system dynamics, positivity constraints and potential system constraints. Below the optimisation problem is given in equation form, see 3-10.

$$\begin{aligned} \min_{\mathbf{u}(k)} \quad & T \cdot \sum_{k=1}^{360} \left\{ \sum_{i=1}^{links} \lambda_i L_i \rho_i(k) + \sum_{j=1}^{ramps} \omega_j(k) \right\}, \ k = 1, \ldots, 360 \\ \text{s.t.} \quad & x(k+1) = x(k) + a(x) \\ & \rho_i(k), v_i(k), \omega_j(k) \geq 0 \\ & 0 \leq u(k) \leq 1 \end{aligned} \tag{3-10}$$

Where $x(k+1) = x(k) + a(k)$ equals the system dynamics as given in the three equations 3-2, 3-3 and 3-4. The positivity constraints make sure the variables for density, speed and the number of cars in the waiting queue cannot become negative, which is of course not realistic. Furthermore, the value for ramp metering and rerouting is in this research set to be discrete en binary, therefore it can only be zero or one. This is done because of a more useful comparison between the NM controller and the MPC controller. The number of links i and ramps j, depend on the network at hand. For network 1 these numbers are 11 links and 3 ramps and for network 2, these are 17 links and 1 ramp.

The resulting optimisation problem is a mixed integer nonlinear programming (MINLP) problem. These are hard problems to solve and without simplification, these problems can be best solved by a heuristic method for instance with a Genetic Algorithm (GA) solver. This will be further explained in the next section 3-5.

In case one wants to solve the problem with another optimisation strategy the problem needs to be rewritten into a mixed integer linear programming (MILP) problem. With the help of piecewise affine (PWA) approximations, this is possible [43]. This approach is out of scope for this thesis, however, this remains an interesting aspect for future research and could potentially improve the performance and help with more complex or larger problems.

Genetic Algorithm

In section 3-1 the reader is introduced to the optimisation algorithm for the testing framework. The MPC controller will be constructed in a Python environment. Therefore an optimisation algorithm suitable for Python is required. For this problem it is chosen to use a genetic algorithm, this is because a heuristic approach was required and typically GA has promising results for these kinds of problems.

```
def optimize(t, state_values, demand, par, mpc, ga, u_current):

    def f(u):
        pen = 0
        TTS, x, u = mpc_prediction(t, state_values, u, demand, par, mpc)

        return TTS + pen

    varbound    = np.array([[0,1]]*mpc.control_variables*mpc.control_horizon)

    algorithm_par = {'max_num_iteration': ga.max_num_iteration,\···

    model = GA(function  = f,
               dimension = mpc.control_variables*mpc.control_horizon,
               variable_type='int',
               variable_boundaries  = varbound,
               algorithm_parameters = algorithm_par,
               function_timeout = 20,
               convergence_curve = False,
               progress_bar = False
               )
    model.run()
    output = model.output_dict
    report = model.report
    return output, report, mpc.control_variables
```

Figure 3-14: Implementation of the PyPi genetic algorithm in Python.

For this research, the GA algorithm, developed by Ryan Solgi [44], is chosen. It is a user-friendly library easy to implement and it had good results. Above one can see the configuration and implementation of the algorithm in Python in figure 3-14. One defines the function $f(u)$ as a function with input $u(k)$ and the return value as the objective function, in this case, the TTS. Additionally one can add penalties on control or state values to the objective function. This function is fed to the algorithm together with its dimensions, variable type & boundaries, algorithm parameters and some extra operational settings. Next, the model can run and the output is the optimal control input calculated by the GA.

```
class ga_settings:
    max_num_iteration = 100
    population_size = 50
    mutation_probability = 0.1
    elit_ratio = 0.01
    crossover_probability = 0.5
    parents_portion = 0.3
    crossover_type = 'uniform'
    convergence_curve ='uniform'
    max_iteration_without_improv = 3
```

(a) Simple freeway network.

```
class ga_settings:
    max_num_iteration = 80
    population_size = 50
    mutation_probability = 0.1
    elit_ratio = 0.01
    crossover_probability = 0.5
    parents_portion = 0.3
    crossover_type = 'uniform'
    convergence_curve ='uniform'
    max_iteration_without_improv = 3
```

(b) Simple freeway plus network.

Figure 3-15: Genetic algorithm parameter settings for network 1 and 2. [Own work]

In figure 3-15 the genetic algorithm parameters can be found for network 1 and network 2. The initial values of the optimisation problem are the first population, in this case, a vector $\mathbf{u}$ consisting of ones and zeros. With the size of the population, the number of initial values is chosen. For network 1 the solution space equals $2^(5 * 3) = 32,768$. The population size is set to 50 and the maximum number of iterations is set to 100. The algorithm can potentially check 5000 options, but it stops after 3 runs without improvement. For network 2 the solution space is equal to $2^{14} = 4,096$. The population size is set to 50 with a maximum number of iterations at 80. The algorithm can potentially check 4000 options, almost all possible solutions, however, after 3 iterations without improvement the algorithm stops and returns the best value so far. The other parameters are kept at the default values.

With this information and number at hand, the use of the genetic algorithm does seem like overkill. Since with these relative small solution spaces, a brute force method, where all possible options are calculated and compared, is possible as well. However, it is the intention to use this framework for larger networks as well. These networks have more control variables and potentially larger control horizons are needed which would increase the complexity of the problem enormously.

Chapter 4

Results and simulations

In this chapter, the results of the simulations will be shared. This will be done for the two networks and the different scenarios. In this chapter, networks 1 and 2 will be discussed. The chapter will start with a focus on the 3 different scenarios as explained in 3-3-2. After that, the scenario where the most impact can be made (the sweet spot of operation), will be further researched with a focus on incorporating penalties and longer on-ramps. In the end, a final simulation will be done with more iterations, including all adjustments and improvements The results will be shared, with more attention to gathering the mean performance over these iterations.

4-1 Results network 1

In this section, the results for scenarios 1, 2 and 3 will be shared for network 1. First, the benchmark results of no control and full control will be shared. The performance will be calculated mostly based on the no control situation, however, it is necessary to check the performance and the behaviour compared to a full control case. If this would be better than all the smart applications, these would have no added value. Nevertheless, it is important to keep in mind that for local segments the full control case could be the best however, control always comes at a price and some vehicles have to be sacrificed. Therefore it is about balancing between control and no control to achieve the best network-wide performance.

Next, the performance of the NM and subsequently the MPC controller will be given for the specific scenarios and compared to each other. Only the most interesting things and segments will be highlighted per scenario through figures, to reduce repetition. Take a look at the configuration in figure 3-12 to understand the numbering of the segments used in this section.

4-1-1 Scenario 1 - below critical density

For the details of scenario 1 take a look at section 3-3-2 in table 3-3. Scenario 1 is the scenario below the critical density, with approximately 10% less demand on the main route.

Benchmarks

Table 4-1: Average benchmarks for scenario 1 regarding network 1 over 5 iterations.

KPI	Value	unit
v_{ffs}	95.81	[km/h]
v_{no}	84.41	[km/h]
v_{full}	65.17	[km/h]
TTS_{no}	289.9	[vehH]

In table 4-1 one can find the benchmarks values for network 1 for scenario 1. The simulation runs for 1 hour. The free-flow speed for network 1 equals 95.81 km/h. The speed of scenario 1 with no control equals 84.17 km/h. There is definitely interaction between the vehicles, such that the speed level drops compared to the free flow case. The full control case is in this case way worse than no control, for the average speed. Therefore the performance will be compared to the no control case. Nevertheless, locally for the on-ramp segments, the full control case will be the best option. However, it is about network-wide performance, for which we evaluate the TTS.

Comparison with NM & MPC

The scores on the main KPI's of the simulations without control, with the NM and with the MPC are given in table 4-2. The improvements in percentages are calculated compared to the no control case and also given.

Table 4-2: Average KPI improvements for scenario 1 regarding network 1 over 5 iterations.

KPI	No control	NM	MPC	unit	Improvement NM	Improvement MPC
v	84.41	84.45	84.07	[km/h]	0.05%	-0.40%
TTS	280.42	280.03	280.75	[vehH]	0.14%	-0.12%

So far no significant performance improvements, however, one can see that both algorithms at least do not make things significantly worse, which is a valuable observation and achievement from the controllers. Next to that, we can observe in figure 4-1, the first decrease in speed for the relatively short segment 6.1.

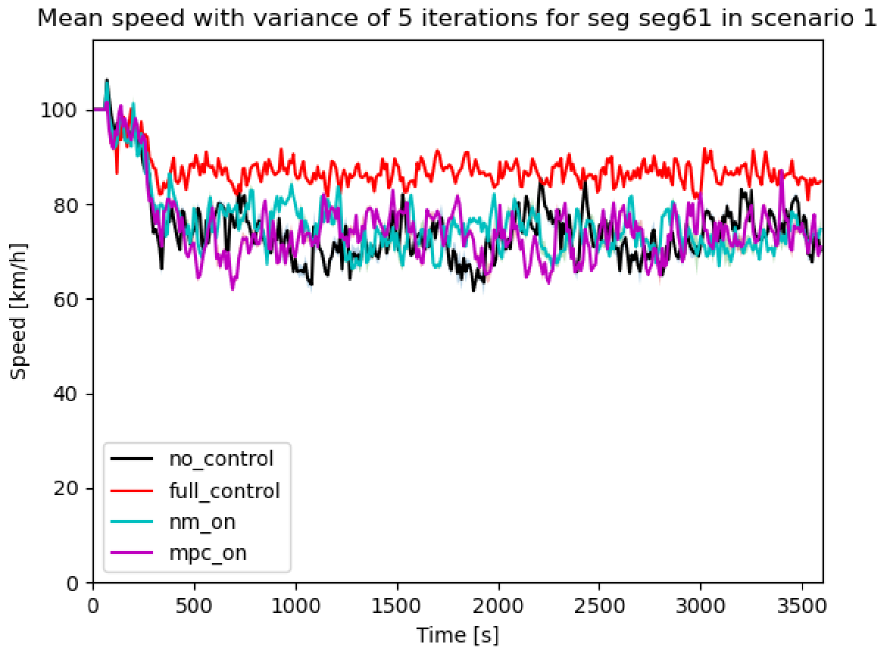


Figure 4-1: Average speed levels of 5 iterations for segment 6.1 in scenario 1 with NM and MPC control, compared to no and full control. The variance is shown by a coloured area filled between the min and max variance.

4-1-2 Scenario 2 - critical density

For the details of scenario 2 take a look at section 3-3-2 in table 3-3. For scenario 2 the demand of the main route is around critical density. The results of the simulations on scenario 2 for network 1 can be found in table 4-3. The speed of scenario 2 with no control equals 69.49 km/h. This is is a small 20% lower than in scenario 1. So there is a clear desire for control and to improve the performance. Compared to scenario 1 the full control case is less bad compared to the no control case, however, it does still make the overall performance worse, looking at the speed values. One can also clearly observe that the value for the TTS has increased compared to scenario 1.

Benchmarks

Table 4-3: Average benchmarks for scenario 2 regarding network 1 over 5 iterations.

KPI	Value	unit
v_{ffs}	95.81	[km/h]
v_{no}	69.49	[km/h]
v_{full}	66.10	[km/h]
TTS_{no}	449.32	[vehH]

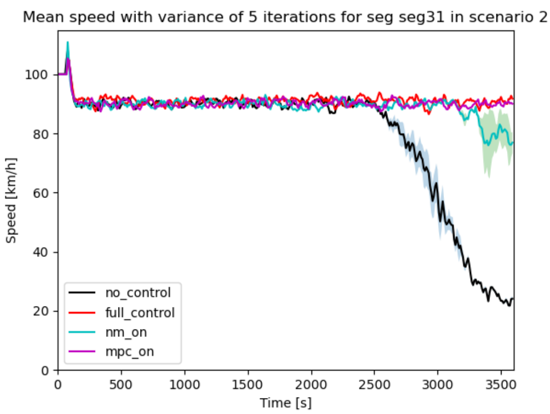
In figure 4-2 and 4-3 the benchmark performance on speed and density of scenario 3 is depicted next to the performance of the control variants for the NM and the MPC controller.

Comparison with NM & MPC

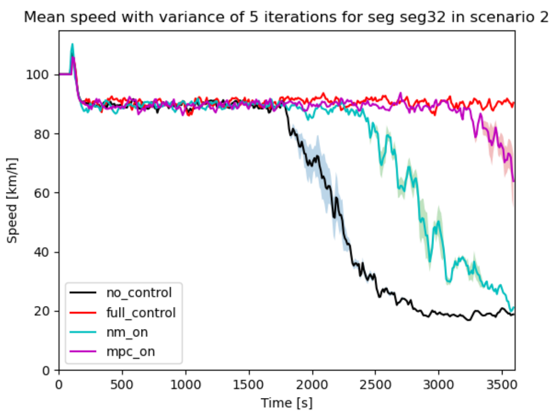
The performance values of scenario 2 concerning the simulations without control and with full control compared to the NM and MPC control are given in table 4-4. We can see an outperformance on the TTS and the speed by the MPC controller compared to the no control case and NM. This is what is expected because the MPC controller optimises the control measures for this performance metric, the TTS.

Table 4-4: Average KPI improvements for scenario 2 regarding network 1 over 5 iterations.

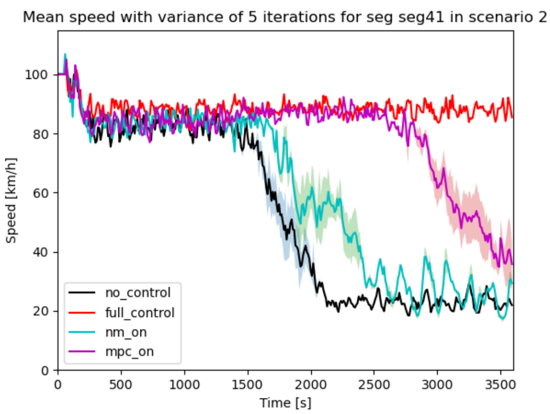
KPI	No control	NM	MPC	unit	Improvement NM	Improvement MPC
v	69.49	70.10	73.76	[km/h]	0.88%	6.14%
TTS	449.32	410.85	386.3	[vehH]	8.56%	14.03%



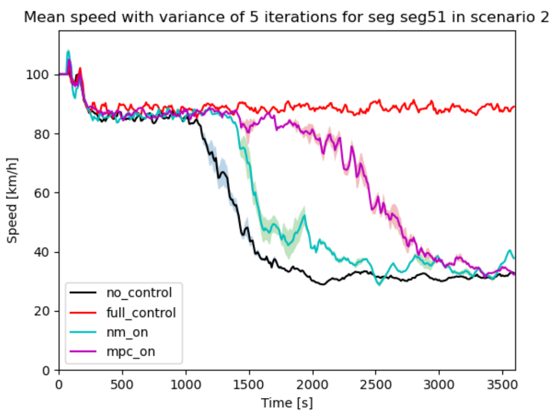
(a) Speed levels segment 3.1



(b) Speed levels segment 3.2



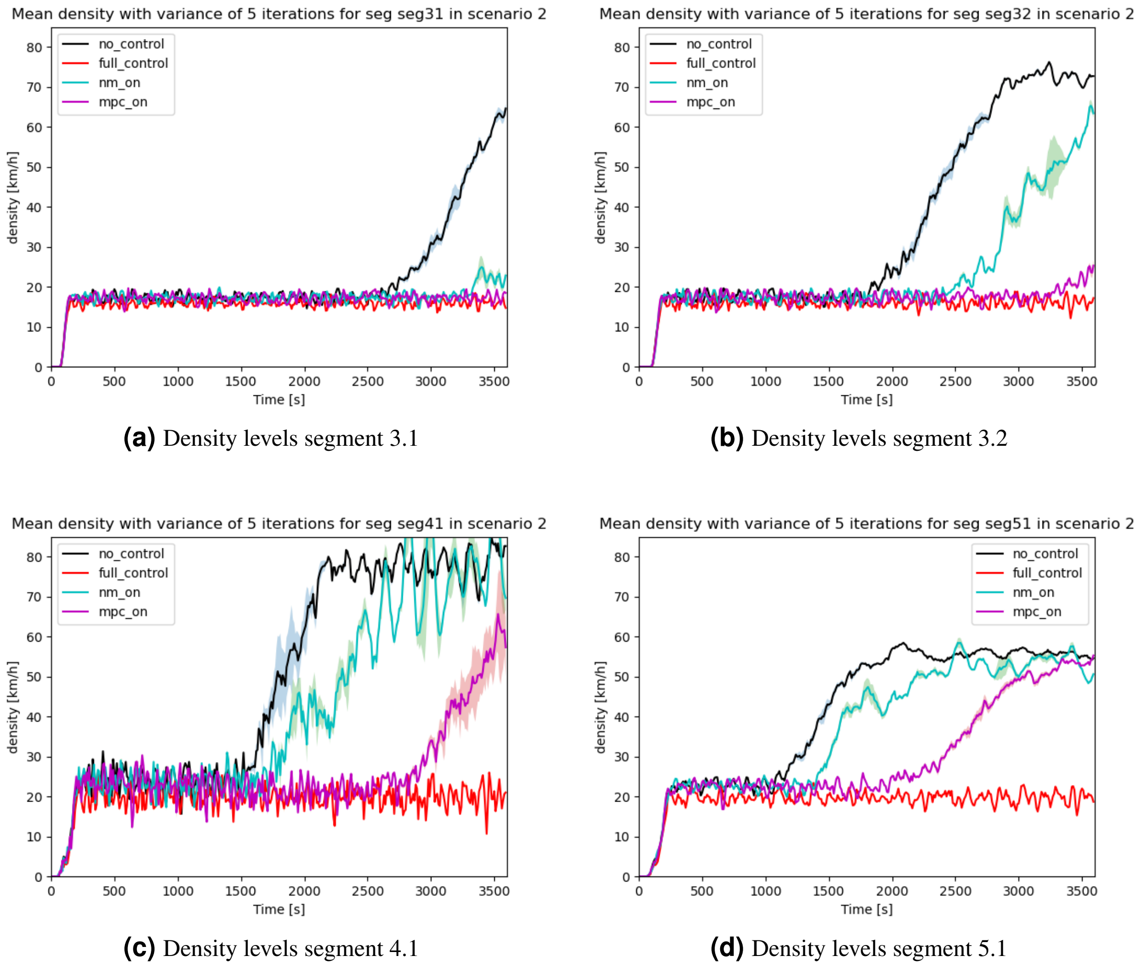
(c) Speed levels segment 4.1



(d) Speed levels segment 5.1

Figure 4-2: Average speed levels for segment 3.1, 3.2, 4.1 and 5.1 in scenario 2 with NM and MPC control, compared to no and full control over 5 iterations. The variance is shown by a coloured area filled between the min and max variance.

In figure 4-2 the speed levels for segment 3.1, 3.2, 4.1 and 5.1 can be seen for scenario 2 with NM control and MPC control compared to no and full control. In this figure, one can observe the clear difference in the performance of the NM and the MPC controller. First of all, one can see the phenomenon that traffic jams grow upstream and move through the network, in case it has no chance to dissolve. For segment 3.1, one can see in figure 4-2a that both controllers can keep the speed for a long time at a high level, comparable to the benchmark speed level in the case of full control. However, in the end, the traffic jam has grown so much that for the case with the NM controller the speed drops at around 3200 seconds. Further in the network, the speed drops already earlier, in figure 4-2d one can see that speed levels drop at around 1000 seconds for the no control case. The NM can manage to postpone the traffic jam and maintain higher speed levels. The MPC controller can even longer continue to maintain higher speed levels.



(a) Density levels segment 3.1

(b) Density levels segment 3.2

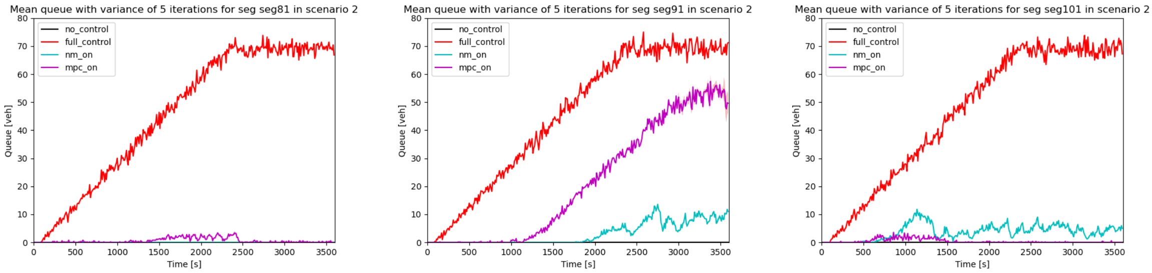
(c) Density levels segment 4.1

(d) Density levels segment 5.1

Figure 4-3: Average density levels for segment 3.1, 3.2, 4.1 and 5.1 in scenario 2 with NM and MPC control, compared to no and full control over 5 iterations. The variance is shown by a coloured area filled between the min and max variance.

Another valuable set of figures is the density levels. See the figures in 4-3. In these figures, one can observe the correlation between the decrease in speed and the increase in density. Furthermore, one can see that the values for segment 4.1 have bigger fluctuations and reach higher values, this is because this segment is only 300 metres and the on-ramp is connected to segment 4.1.

Lastly is important to show that everything comes at a price. The segments where the controllers make things worse are on the on-ramps of course. This can be noted in the figures in 4-4, there is a clear difference with the full control case. The max of the queue is around 70 cars, this is due to the length of the segment, which was configured this way. In a later section on penalties, section 4-3 the effects of an even longer on-ramp will be tested, in the hope of better MPC performance.



(a) Queue level segment 8.1 with NM

(b) Queue level segment 9.1 with NM

(c) Queue level segment 10.1 with MPC

Figure 4-4: Average queue levels for on-ramps 8.1, 9.1 and 10.1 in scenario 2 with NM and MPC control, compared to no control. The variance is shown by a coloured area filled between the min and max variance.

4-1-3 Scenario 3 - above critical density

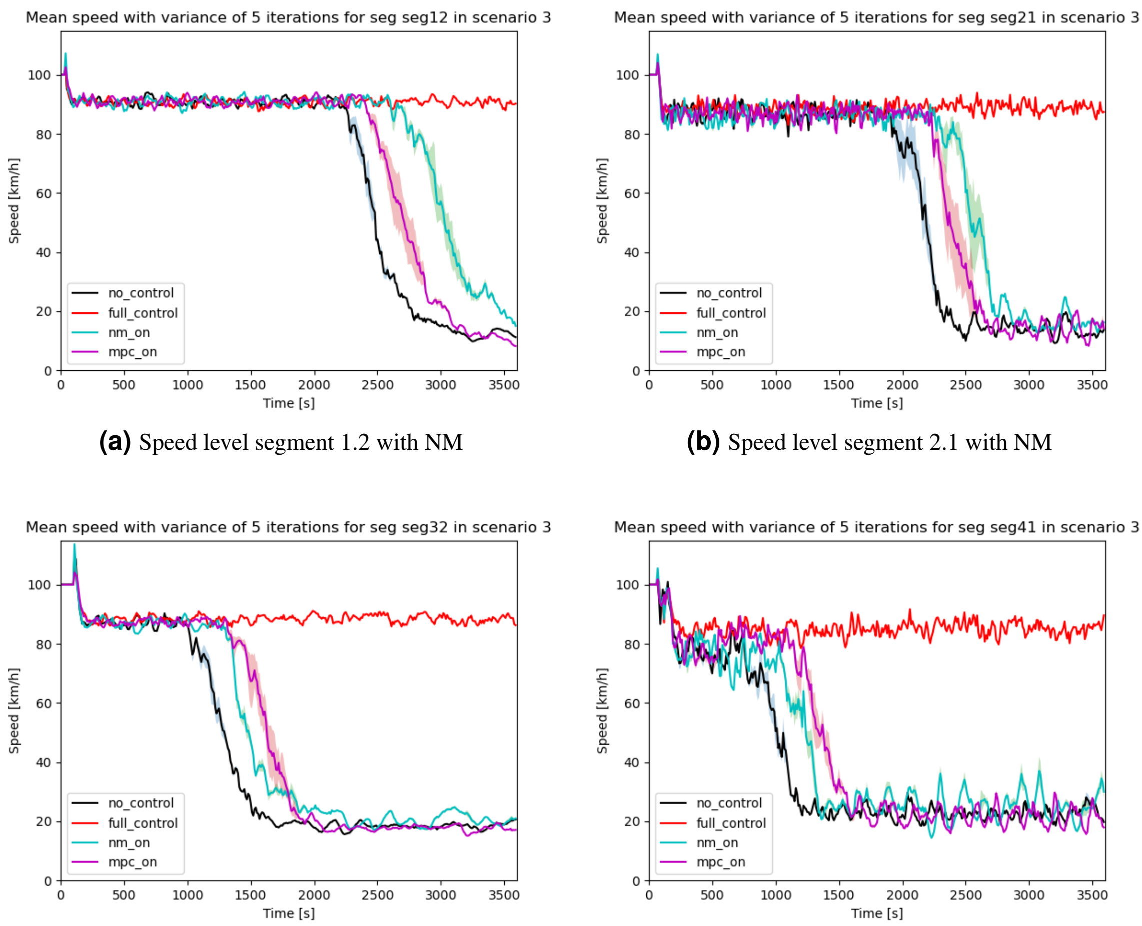
For the details of scenario 3 take a look at section 3-3-2 in table 3-3. For scenario 3 the demand of the main route is approximately 10 % above critical density. In table 4-5 the benchmarks values for network 1 for scenario 3 are given. The speed of scenario 3 with no control equals 62.76 km/h. This is a lower speed with a little more than 10% compared to scenario 2. The control options are still of value, taking a look at the full control case confirms this. The speed in the full control case is equal to 66.20 km/h. The TTS does increase again significantly compared to scenario 3, also due to more cars this is completely logical. So as expected and given in table 3-5 the overall network performance is worse, not meaning there are no improvements possible.

Benchmarks

Table 4-5: Average benchmarks for scenario 3 regarding network 1 over 5 iterations.

KPI	Value	unit
v_{ffs}	95.81	[km/h]
v_{no}	62.76	[km/h]
v_{full}	66.20	[km/h]
TTS_{no}	606.22	[vehH]

In figure 4-5 the benchmark performance on the speed of scenario 3 is depicted next to the performance of the control variants for the NM and the MPC controller.



(a) Speed level segment 1.2 with NM

(b) Speed level segment 2.1 with NM

(c) Speed level segment 3.2 with MPC

(d) Speed level segment 4.1 with MPC

Figure 4-5: Average speed levels for segment 1.2, 2.1, 3.2 and 4.1 in scenario 3 with NM and MPC control, compared to no and full control over 5 iterations. The variance is shown by a coloured area filled between the min and max variance.

Comparison with NM & MPC

In the table 4-6 the performance of the controllers in scenario 3 can be found, the improvement percentages are given compared to the no control case. The NM scores 4% higher on TTS improvement compared to the MPC case.

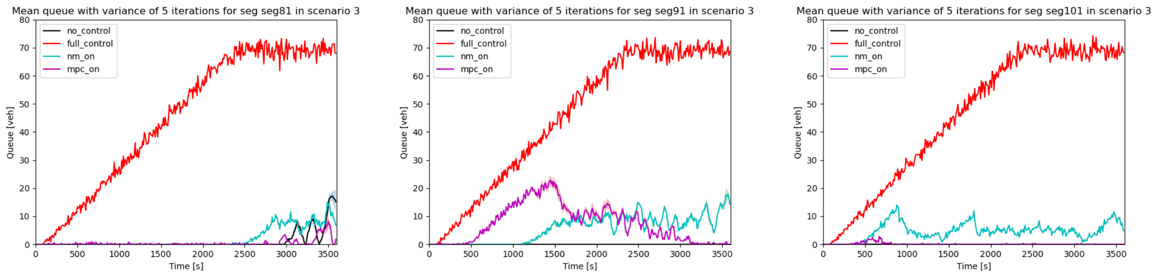
Table 4-6: Average KPI improvements for scenario 3 regarding network 1 over 5 iterations.

KPI	No control	NM	MPC	unit	Improvement NM	Improvement MPC
v	62.76	62.33	63.77	[km/h]	-0.69%	1.61%
TTS	606.22	554.28	581.02	[vehH]	8.57%	4.16%

For this scenario, where the traffic is really dense, something remarkable happens. Take a look at figure 4-5.

For segments 3.2 and 4.1 one can observe that the MPC controller will act earlier and keep the speed level a little higher. However, for segments 1.2 and 2.1 the speed levels are kept longer at a high level by the NM. A possible explanation for the underperformance of the MPC controller could be a mismatch in the model configuration. One could at least cautiously observe that for higher density the NM outperforms the MPC controller.

The difference in queue length is also interesting to observe and can be found in figure 4-6.



(a) Queue level segment 8.1 with NM

(b) Queue level segment 9.1 with NM

(c) Queue level segment 10.1 with MPC

Figure 4-6: Average queue levels for on-ramps 8.1, 9.1 and 10.1 in scenario 3 with NM and MPC control, compared to no control over 5 iterations. The variance is shown by a coloured area filled between the min and max variance.

Remarkably, the performance is different from scenario 2. Looking at for example the ramp metering control and thus queue length for ramp segment 9.1 in figure 4-4b and 4-6b. The model and MPC controller seem to be not working well for the more dense scenario, while NM still improves the TTS for the entire network by 8,6%.

4-2 Results network 2

In this section, the results for scenarios 1, 2 and 3 will be shared for network 2. First, the benchmark results of no control and full control will be shared. The performance will be calculated mostly based on the no control situation, however, it is necessary to check the performance and the behaviour compared to a full control case. If this would be better all the smart applications have no added value. Nevertheless, it is important to keep in mind that for local segments the full control case could be the best however, control always comes at a price and some vehicles have to be sacrificed. Therefore it is about balancing between control and no control to achieve the best network-wide performance.

Only the most interesting things and segments will be highlighted per scenario using figures, to reduce repetition. Take a look at the configuration in figure 3-13 to understand the numbering of the segments used in this section.

4-2-1 Scenario 1 - below critical density

For the details of scenario 1 take a look at section 3-3-2 in table 3-3. For scenario 1 the demand of the main route is approximately 10% below critical density. In table 4-7 one can find the benchmarks values for network 2 with scenario 1. The simulation runs for 1 hour. The free-flow speed for network 1 equals 97.23 km/h.

Benchmarks

Table 4-7: Average benchmarks for scenario 1 regarding network 2 over 5 iterations.

KPI	Value	unit
v_{ffs}	97.23	[km/h]
v_{no}	79.42	[km/h]
v_{full}	77.28	[km/h]
TTS_{no}	948.14	[vehH]

The average speed of scenario 1 with no control equals 79.42 km/h so there is definitely interaction between the vehicles, such that the speed level drops compared to the free-flow speed. However critical densities are not reached, only for segment 2.1 where the on-ramp is connected to and segment 1.6 just before, see 4-7a and look for the no control case.

In this case, the full control approach is a little worse than no control based on the average speed. Therefore the performance will be compared to the no control case. What can already be concluded from this scenario is that rerouting causes extra vehicle interactions at segment 0.1 leading to speed drops, see figure 4-7b and take a look at the full control case. This negative effect will increase with higher densities.

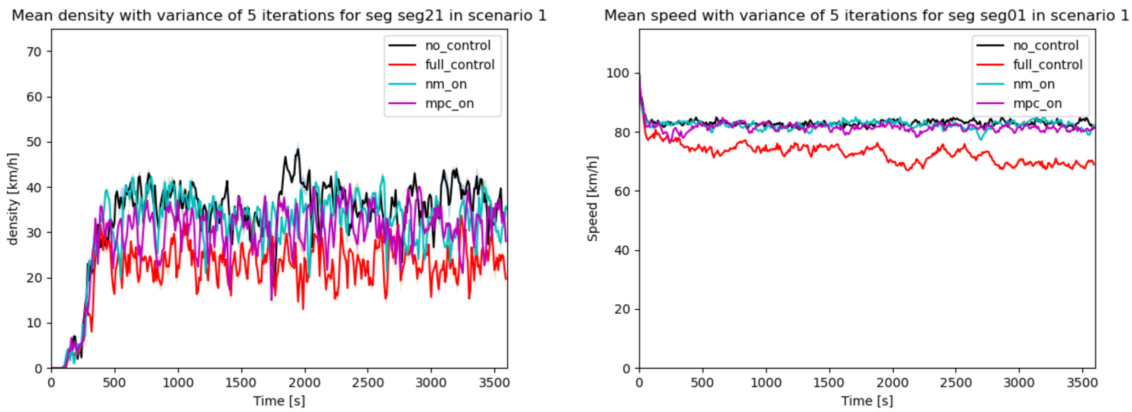
Comparison with NM & MPC

In table 4-8 the overall performance of the controllers in scenario 1 can be found compared to the no control case. Comparable small improvements are achieved by both controllers.

Table 4-8: Average KPI improvements for scenario 1 regarding network 2 over 5 iterations.

KPI	No control	NM	MPC	unit	Improvement NM	Improvement MPC
v	79.42	79.50	79.47	[km/h]	0.10%	0.06%
TTS	948.14	947.68	948.02	[vehH]	0.05%	0.01%

The speed and density profiles are comparable for the NM and MPC control cases. The differences are made mainly for segment 2.1. However, the effects of segment 2.1 are really low compared to the entire network. The results of the NM controller and the MPC controller can be found in figure 4-7.



(a) Density levels segment 0.1 **(b)** Speed levels segment 0.1

Figure 4-7: Average density and speed levels over 5 iterations for segment 0.1 and 1.6 in scenario 1 with NM and MPC control compared to no and full control. The variance is shown by a coloured area filled between the min and max variance.

4-2-2 Scenario 2 - critical density

For the details of scenario 2 take a look at section 3-3-2 in table 3-4. For scenario 2 the demand of the main route is approximately around critical density.

Benchmarks

Table 4-9: Average benchmarks for scenario 2 regarding network 2 over 5 iterations.

KPI	Value	unit
v_{ffs}	97.23	[km/h]
v_{no}	73.38	[km/h]
v_{full}	75.95	[km/h]
TTS_{no}	1155.63	[vehH]

In table 4-9 the benchmarks values for network 2 for scenario 2 can be found. The speed of scenario 2 with no control equals 73.38 km/h. This is is a small 8% lower than in scenario 1. So there is a potential desire for control and improved performance. Another remarkable fact is that the full control case leads in this case even to a higher average speed compared to the no control case, this is because always rerouting has a major positive effect on the average speed and TTS. Interactions with inserting cars are the biggest cause of traffic jams, definitely with higher already higher densities on the mainstream.

Comparison with NM & MPC

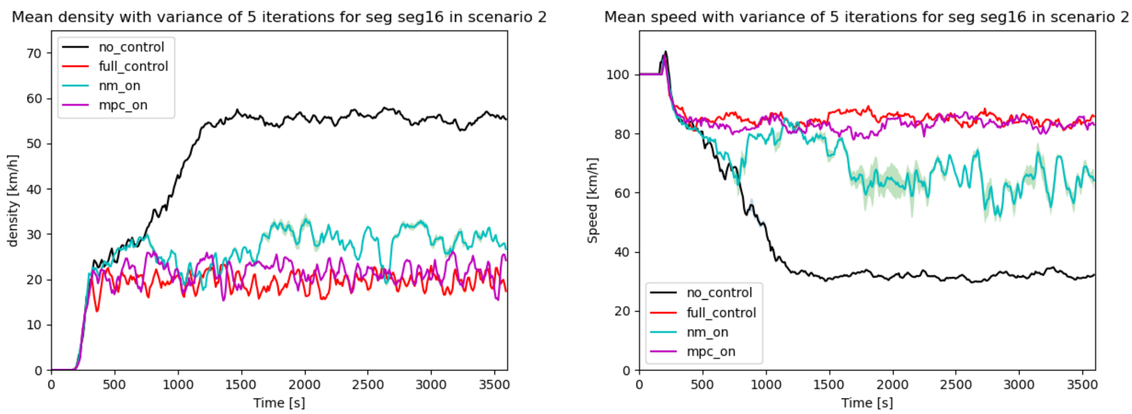
In table 4-10 the overall performance of the controllers in scenario 2 can be found, where they are compared to the no control case.

Table 4-10: Average KPI improvements for scenario 2 regarding network 2 compared to no control over 5 iterations.

KPI	No control	NM	MPC	unit	Improvement NM	Improvement MPC
v	73.38	75.84	76.99	[km/h]	3.35%	4.92%
TTS	1155.63	1034.89	1017.21	[vehH]	10.45%	11.98%

Here are significant performance improvements made of more than 10% for the TTS. However, compared to the full control case numbers are less good. Due to the effect explained in the previous section. Next to that, there is also the problem that some cars on the on-ramp are not stored in the queue variable w, because the max number of cars is reached at 2300 seconds. Therefore the TTS number for the full control case is slightly more positive than it should be, this is the case for all the scenarios. This effect will be accounted for in the final simulation runs in section 4-4, with a longer on-ramp.

The improved performance can graphically best be seen for segment 1.6 in figure 4-8.



(a) Density levels segment 1.6

(b) Speed levels segment 1.6

Figure 4-8: Average density and speed levels over 5 iterations for segment 1.6 in scenario 2 with NM and MPC control compared to no and full control. The variance is shown by a coloured area filled between the min and max variance.

4-2-3 Scenario 3 - above critical density

For the details of scenario 3 take a look at section 3-3-2 in table 3-4. For scenario 3 the demand of the main route is approximately 10% above the theoretical critical density, increasing the flow of route 1 from 4000 veh/hour to 4400 veh/hour.

Benchmarks

Table 4-11: Average benchmarks for scenario 3 regarding network 2 over 5 iterations.

KPI	Value	unit
v_{ffs}	97.23	[km/h]
v_{no}	72.08	[km/h]
v_{full}	75.73	[km/h]
TTS_{no}	1243.77	[vehH]

In table 4-11 the benchmarks values for network 2 for scenario 3 can be found. The TTS increases by 7.6% compared to scenario 2. The speed of scenario 3 with no control equals 72.08 km/h and with full control equals 75.73. Inducing that full control is a good alternative compared to no control. Certainly, because there is an even extra off-balance between the two routes, which can be minimised by the rerouting control measure.

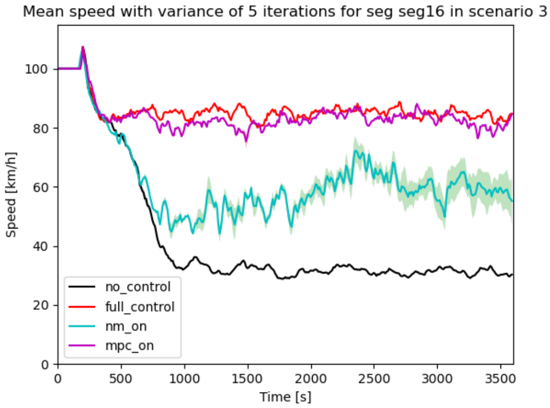
Comparison with NM & MPC

In table 4-12 the overall performance of the controllers in scenario 3 can be found compared to the no control case.

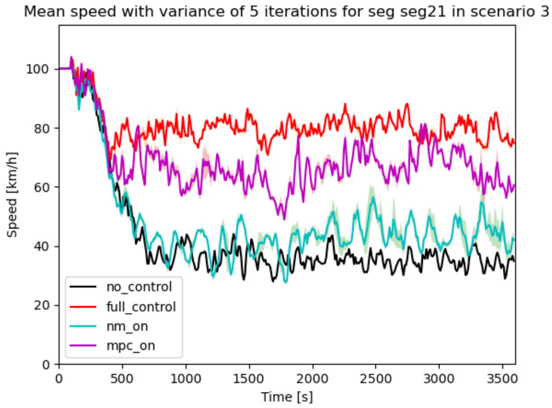
Table 4-12: Average KPI improvements for scenario 3 regarding network 2 compared to no control over 5 iterations.

KPI	No control	NM	MPC	unit	Improvement NM	Improvement MPC
v	72.08	74.83	77.16	[km/h]	3.82%	7.05%
TTS	1243.77	1052.21	1015.12	[vehH]	15.40%	18.38%

One can observe high improvement scores for both the control cases. The main difference between the MPC and NM controller are made on the segments 1.6 and 2.1, see figure 4-9.



(a) Speed levels segment 1.6



(b) Speed levels segment 2.1

Figure 4-9: Average density and speed levels with variance over 5 iterations for segment 2.1 in scenario 3 with NM and MPC control compared to no and full control. The variance is shown by a coloured area filled between the min and max variance.

4-3 Penalties

In this section, we will discuss the possibility to work with penalties with the MPC controller. The use of penalties could be very beneficial to achieve some desired performance or to diminish undesired behaviour. Two possible penalties are discussed. The first is a penalty for switching the control setting, to diminish noisy behaviour. The second is a penalty for a certain queue length, to show it is possible to keep the queue length in control. The penalties will be added to the cost function 3-9. First, the penalties will be implemented and the process will be explained. Next, the results will be shared.

4-3-1 Switching control penalty

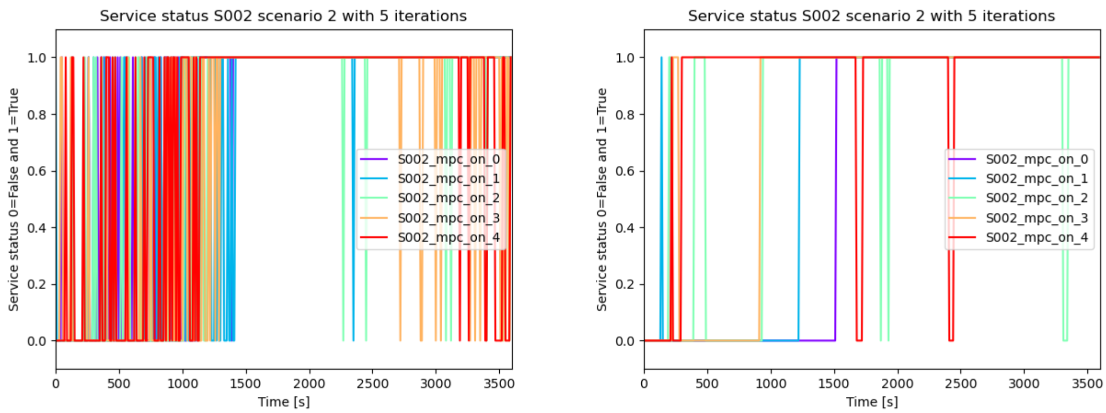
To test this penalty in the simulation environment, scenario 2 is taken for network 1 and scenario 3 for network 2. These were the scenarios where the MPC controller had the most impact and improvement, given the results in section 4-1 and 4-2.

The penalty for the switching control setting is calculated by multiplying the absolute value of the difference with a certain α. Where the difference is taken by, subtracting the first value of each future control variable of the GA possible solution, with the current value of the control variable. The height of the α has been determined in a heuristic manner, where multiple weights have been considered.

In this way, the GA is more likely to suggest a switch only if it has sufficient or significant improvement. The call on which α leads to the desired performance has been done by balancing between performance percentages and observing fewer switches in the control setting.

Network 1

Below in figure 4-10 one can observe the difference in control activation for the second ramp meter in network 1 with an α equal to 0.0025. On the left side we see definitely more fluctuations between on and off than for the case with penalty.



(a) Service status no penalty

(b) Service status with penalty

Figure 4-10: Service status ramp meter 2 without and with penalty for switching cost, over 5 iterations for network 1 with scenario 2.

See table 4-13 for the results on the TTS for the simulation with penalty for switching control compared to the no control case.

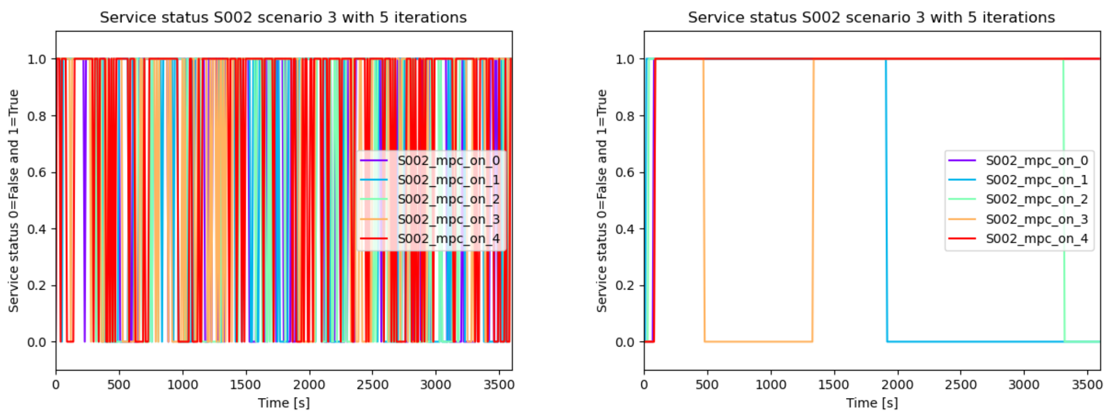
Table 4-13: Average TTS for network 1 with scenario 2 over 5 iterations with penalty for switching control.

Network	α	NM	MPC
1	0.0025	6.08%	8.03%

The rather low α can be logically explained since it is about the difference a control activation or not, can make in only one time-step equal to 10s. This is a small part of the entire prediction horizon for which the TTS is calculated.

Network 2

Below in figure 4-11 one can observe the difference in control activation for the reroute control in network 2 with an α equal to 0.005. On the left side we see definitely more fluctuations between on and off than for the case with penalty.



(a) Service status no penalty **(b)** Service status with penalty

Figure 4-11: Service status reroute without and with penalty for switching cost, over 5 iterations for network 2 with scenario 3.

See table 4-14 for the results on the TTS for the simulation with penalty for switching control compared to the no control case.

Table 4-14: Average TTS compared to no control for network 2 with scenario 3 over 5 iterations with penalty for switching control.

Network	α	NM	MPC
1	0.005	4.28%	7.04%

4-3-2 Queue length penalty

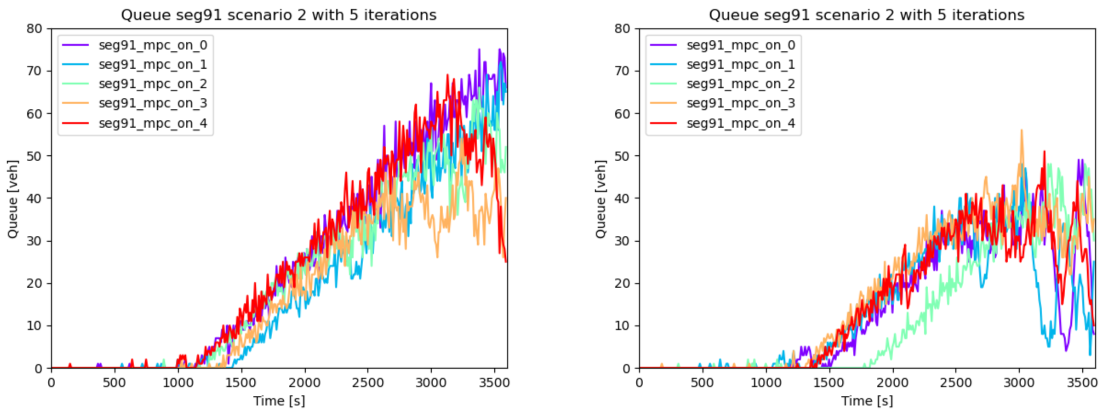
To test this penalty in the simulation environment we only take a look at network 1 with scenario 2. This has been the only scenario where the queue length grew, for the full-control and MPC-on case. Next to that, this was the scenario where the MPC controller had the most impact and improvement, given the results in section 4-1 and 4-2.

This penalty is meant for penalising queues that exceed a certain threshold value β. When simulating all time-steps within the prediction horizon, for a candidate solution of the GA, the penalty is calculated cumulatively, by adding the sum of $\beta \times \gamma$. Where γ is the penalty variable that is heuristically determined. For which multiple weights have been considered. The final penalty is added to the cost function, TTS.

Activating this penalty would desirably lead to queues that are not likely to exceed 50 cars, at most incidentally.

Network 1

Below in figure 4-12 the results are shown of 5 iterations of simulation with a β equal to 50 and γ equal to 0.0025. One can clearly observe the desired performance between the two figures 4-12a and 4-12b.



(a) Queue length no penalty

(b) Queue length with penalty

Figure 4-12: Queue length for segment 9.1 without and with penalty, over 5 iterations for network 1 with scenario 2.

See table 4-15 for the results on the TTS for the simulation with penalty for exceeding a threshold queue length value, compared to the no control case.

Table 4-15: Average TTS for network 1 with scenario 2 over 5 iterations with penalty for exceeding threshold queue length.

Network	β	γ	NM	MPC
1	50	0.0025	7.31%	13.35%

4-4 Final results

In this section, the results for scenario 2 for both networks will be tested again in a more sophisticated way. For this simulation run, 20 iterations have been done. This will give more weight to the conclusions and final numbers. Besides that, it verifies that the controllers are not just performing incidentally. Note, that the on-ramp lengths are enlarged to have a more representative TTS for the full control case. The penalties are not activated. The simulation time per iteration was again 1 hour. Scenario 2 is chosen for network 1 because this is seen as the sweet spot of operation, where the controllers can make the most impact. Also for network 2 scenario 2 is chosen instead of 3, because in this scenario it is always lucrative to reroute, because of the off-balance between demand for routes 1 and 2.

4-4-1 Network 1

Below in table 4-16 one can find the results for the final simulation run with 20 iteration for network 1 for scenario 2. Both controllers perform well and achieve a significant improvement for the Total Time Spent (TTS) in network 1. The MPC controller outperforms the NM.

Table 4-16: Average KPI improvements for scenario 2 regarding network 1 compared to no control over 20 iterations.

KPI	No control	NM	MPC	unit	Improvement NM	Improvement MPC
v	72.16	72.04	73,45	[km/h]	-0.17%	1.79%
TTS	407.31	386.97	377.53	[vehH]	4.99%	7.31%

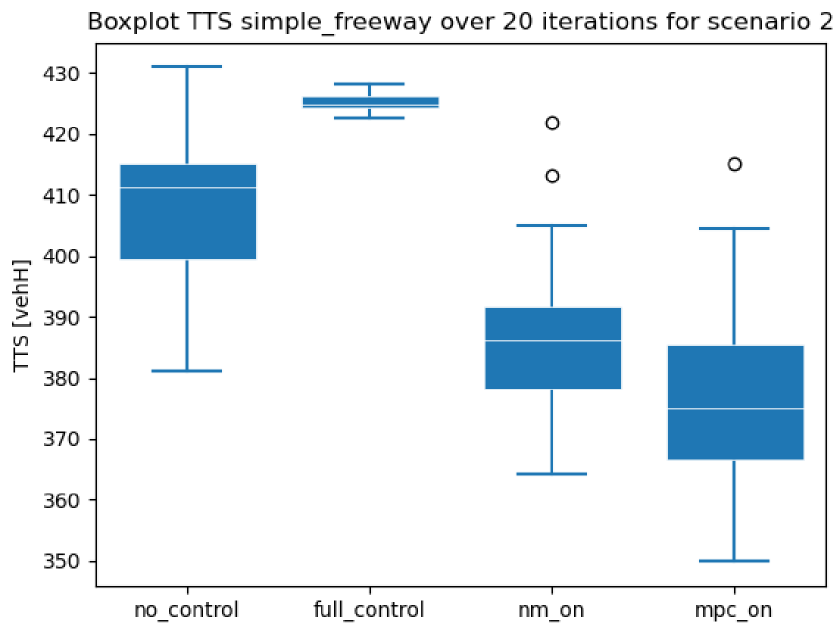


Figure 4-13: Box plot of TTS of 20 iteration for network 1 with scenario 2.

To show the dispersion between the different iterations and the different control cases, see the box-plot of the TTS above in figure 4-13. It is also clear that full control is in this case not the better option and makes the TTS worse compared to no control.

4-4-2 Network 2

Below in table 4-17 one can find the results for the final simulation run with 20 iteration for network 2 for scenario 2. Both controllers perform well and achieve a significant improvement for the TTS in network 2. The MPC controller slightly outperforms the NM.

Table 4-17: Average KPI improvements for scenario 2 regarding network 2 compared to no control over 20 iterations.

KPI	No control	NM	MPC	unit	Improvement NM	Improvement MPC
v	73.56	75.94	76.84	[km/h]	3.24%	4.46%
TTS	1139.16	1030.29	1014.66	[vehH]	9.56%	10.93%

To show the dispersion between the different iterations and the different control cases, see the box-plot of the TTS below in figure 4-14.

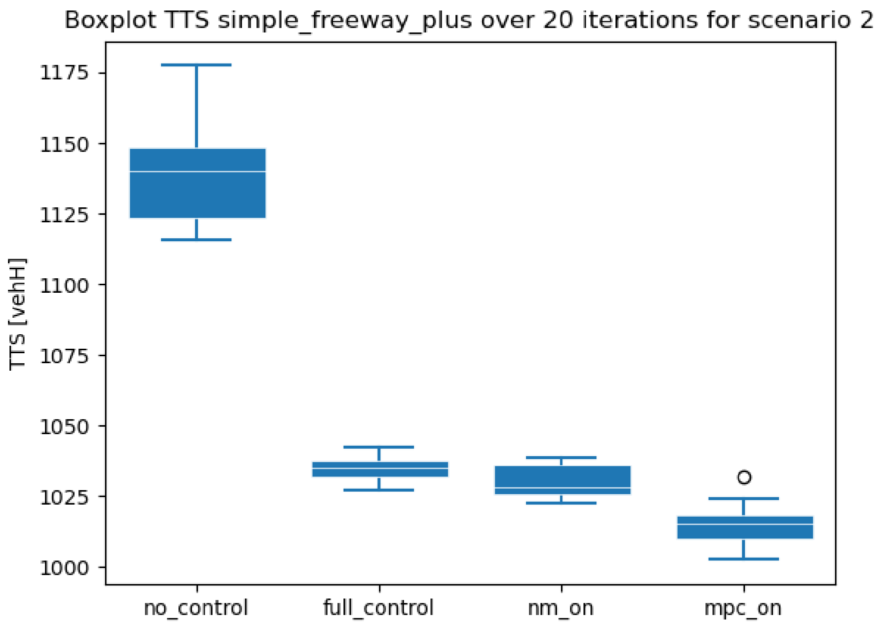


Figure 4-14: Box plot of TTS of 20 iteration for network 2 with scenario 2.

It can be observed that the full control case also performs better than the no control case. So even for scenario 2, it is lucrative to always reroute. The little extra performance the NM and MPC controller achieve is the difference in control settings for the on-ramp, which has for this network respectively little impact. If one would increase the flow for the on-ramp or the amount of on-ramp the full control case would underperform more.

The effect of the demand balance between routes 1 and 2 can also be noticed when comparing the performance of the final simulation run to the one in the previous section. The TTS for full control equals 990.55 and for no control equals 948.14. In this case, always rerouting is not lucrative anymore.

Chapter 5

Conclusions

For this thesis, a testing framework for multiple control algorithms and traffic networks is proposed. Multiple simulations are done for two different networks. Each network is tested for different scenarios and four different control approaches. With the help of this framework, we attempted to answer the question: "Can a user-friendly, transparent, network-wide traffic control algorithm achieve a promising performance compared to more advanced, although non-transparent control methods for large mixed city and freeway networks?" The results support a 'yes'. However, also some notes must be made. In the next section, a summary of the results, conclusions and side notes will be given. Lastly, some possible applications will be shared.

5-1 Summary of results

In the final simulation run for network 1, with 20 iterations, both algorithms achieved significant improvement. The Network Manager (NM) improved the Total Time Spent (TTS) by 5% and the Model Predictive Control (MPC) controller by 7.3%. For network 2 the NM improved the TTS by 9.6% and the MPC controller by 10.9%. Underlining the use of network-wide traffic control. Furthermore, this shows that the NM can achieve a comparable, yet little worse performance compared to an advanced control method like MPC for (large) freeway traffic networks.

A look at the results of scenario 2 for network 1 in figure 4-2 suggests that a reason for the underperformance of the NM in certain cases is due to the lack of a prediction element. During the development of the NM, Technolution was also planning to incorporate this into the algorithm. Technolution is advised to develop this for the NM, given that the development of an MPC controller turns out to be too complex.

Looking at the results for the first segment of network 2, we can conclude that rerouting also harms the speed and thereby time loss due to forced lane changes by control. To what extent this is also true for a real case scenario is hard to say. It could be that the lane changes in the simulation are too abrupt and therefore should be done in a shorter amount of time and space compared to real-life situations.

The fact that everything comes at a price can also be derived from the results. The potential improvements that can and are made, harm other road segments. This will always be the case. However, we can see that network-wide performance improvement can be achieved by sacrificing certain network users. Furthermore, the tests also show that by using penalties, the sacrificing can be limited to what is acceptable by tuning the penalty variables for specific cases, while still obtaining overall performance improvement.

In some cases, MPC can also underperform compared to full-control. This depends on the demand and scenario configuration. Scenarios 2 and 3 for network 2 are now configured for a more static one-way control approach, where it is almost always better to reroute the 10% to prevent the interaction with the on-ramp. Therefore, we could conclude that in case the off-balance between the two different routes is around 10% it is probably better to activate rerouting.

For network 2 the results showed that a single ramp meter has little influence on the TTS for a larger network. Based on these results for network 2 it can be concluded that the larger the network becomes, the larger the TTS becomes and the less a single control action influences this TTS. This does not necessarily have to be a problem. Municipalities could for instance also give priority to certain road segments by using weights to calculate the TTS. In this way, certain segments or control influences could be enhanced.

One of the sub-question for this research was: "How suitable is the MPC controller approach in case the network becomes more complex?" There will be more challenges and complications when the networks become more complex. With the results and knowledge so far, it is believed there is certainly potential for MPC, nevertheless, further research is required. This will be shared with you in the next chapter in section 6-2.

5-2 Applications

One of the applications of the framework is to support companies in developing, testing and analysing current and new algorithms. Where companies could use the framework in different environments and use different settings. Another application of this thesis for Technolution could be to include the MPC approach in their operation. However, still many things need to be taken into account. For example, whether the data that is required, is available. One would need the average speeds, densities and waiting queues. Most of these can be collected with contemporary sensors. Another aspect to consider is the configuration of more complex traffic controls instead of the simpler ramp metering and rerouting features. At last, the NM could be improved with the findings of this research. Close cooperation with the municipalities, where the NM is operational now, is needed.

5-3 Goals and objectives

With this research, the main goal of building a test framework is accomplished. Technolution could use the framework for further research, different use-cases and scenarios or implement a new algorithm. For the goal, give direction for further development, it is believed further research is required. As for the objectives, most of them are realised. City traffic has unfortunately been out of scope for this research during the practical implementation, therefore this part of the research question could not be answered. Nevertheless, interesting papers have been found and studied which would give good direction for further research [8], [9], [45], [20], [34], [46], [47].

Chapter 6

Recommendations

During this research, a lot of different simulations have been done throughout the entire process. This led to useful knowledge and know-how in a lot of aspects. Multiple ideas have come to mind for future work and recommendations for the current work. In this chapter, the most relevant improvements, adjustments and future work recommendations will be highlighted.

6-1 Improvements and adjustments

6-1-1 Demand & scenario's

Now the fundamentals of the framework are ready, it is interesting to test for multiple traffic specific scenarios. For instance, accidents could be simulated, or the demand scenarios could be configured with temporarily higher demands instead of continuous higher demand. Working with intervals for the demand is supported by SUMO. A third option could be to increase the demand gradually with these intervals. MPC will become less accurate because it has only 1 known inflow number for the METANET model. However, it has the advantage of the prediction compared to the NM by growing density. One could also make the model smarter if rush hour demands are known for example. However, that could also result in more inaccuracies.

6-1-2 Professionalise framework

In case the framework is going to be used internally for more tests or the framework is used for future research, it is advised to code a built-in control test to be 100% sure of the correctness all the time. This is certainly necessary when people start tuning and changing settings. The larger the network becomes, the easier small configuration mistakes occur. Many configuration files need to be adjusted with care and structure. New people will make these mistakes sooner. Another example of a certain test could be to check for the actual flow in the network and compare this to the given flow in the demand configuration files. Possibly set demands cannot be reached due to limits on the insertion capacity of SUMO and the network, see [48].

6-1-3 METANET improvements

The METANET model is a model with a long history. This means, that already a lot of adjustments and improvements have been done over time. Extra improvements in addition to the current model could be to add extra speed terms accounting for speed drops due to merging phenomena and speed reduction due to weaving phenomena, see [3].
In case the application would be further investigated with the use of MPC, the METANET model parameters could be adjusted to a specific network configuration. The current model parameters are taken from the literature. However, if one would like to exploit the application commercially, a certain city or municipality will be chosen. If the data would be available, the parameters could be acquired by doing a data regression analysis [49].

6-1-4 Control configuration

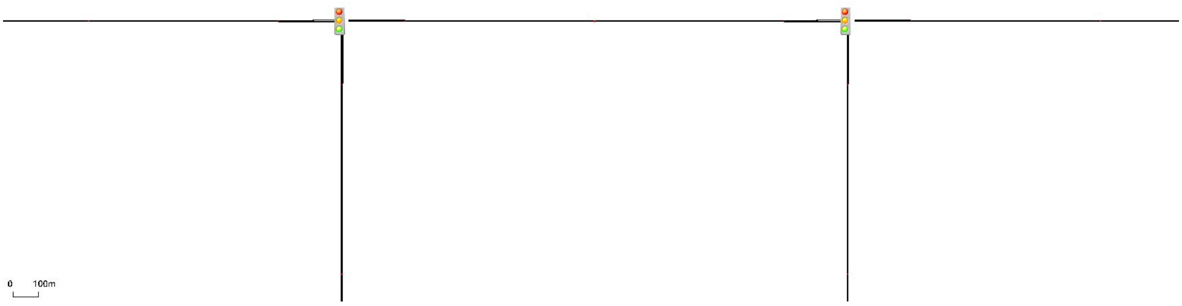
Other useful changes and adjustments that are recommended, are control related. For instance, changing the effect of the on-ramp. At this point, the ramp meters in the research are configured as flickering traffic lights with constant and static settings. This implies, 2 seconds red time, 2.10 seconds yellow time and 2 seconds green time. These settings result in a 0.3 pass rate for the cars when ramp metering control is active.
The other control measure is rerouting. At this point, rerouting is configured as a measure using Variable Message Signs (VMS), which typically has a low follow-up rate. In this thesis the rate is set to a 10% change to reroute a car when rerouting control is active. One could also simulate rerouting more as an in-car advice control measure, which normally has a way higher follow-up rate.
Finally, the control figuration of the network manager itself could also be tested further. Either by changing the trigger areas and for example, ensuring no overlap in the case of network 2, or by testing for shorter or longer trigger areas for network 1.

6-1-5 Optimisation algorithm

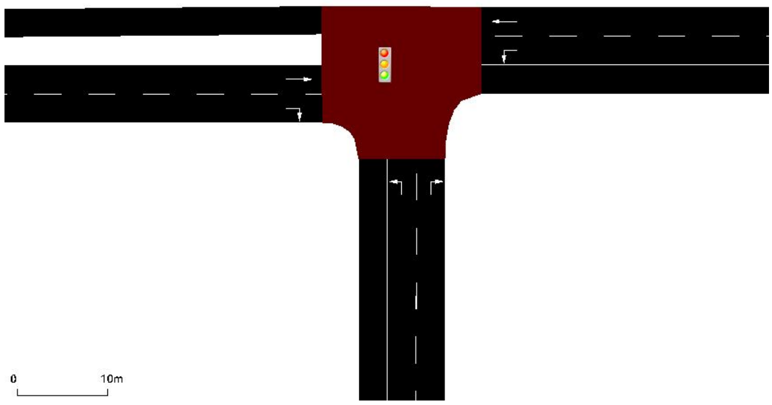
It could be promising to change the optimisation algorithm to improve performance and speed. The latter would be of extra interest in case larger networks are considered and the problem intensifies. To make sure a non-heuristic approach can be used, one will have to dive into rewriting the optimisation problem with the help of a Piecewise Affine (PWA) approximation. For instance, as is done in [43].

6-2 Future work

A desirable next step for future work would be to investigate the performance of both control approaches for a larger network. A network with multiple services that differ in control type actions. For instance, somewhere in the range of 40 to 50 services. Next to that, it could also be interesting to see the performance on such a larger network, that it is suitable for city and freeway traffic. This requires the implementation of an additional model for the MPC controller. Papers like [8], [9], [45], can be used as reference and starting point. In these papers, they combine the METANET and an extended urban model based on [46] to a macroscopic model that is suitable for an MPC approach. For the sake of the proposed further research, it would be useful to start with a small network just suitable for city traffic. For instance, the Pi-network which is depicted below in figure 6-1. Before one would jump into the larger network. The knowledge of both could then be combined for a larger network. Something like the ring-network below, see figure 6-2. With perhaps a little more city traffic roads.



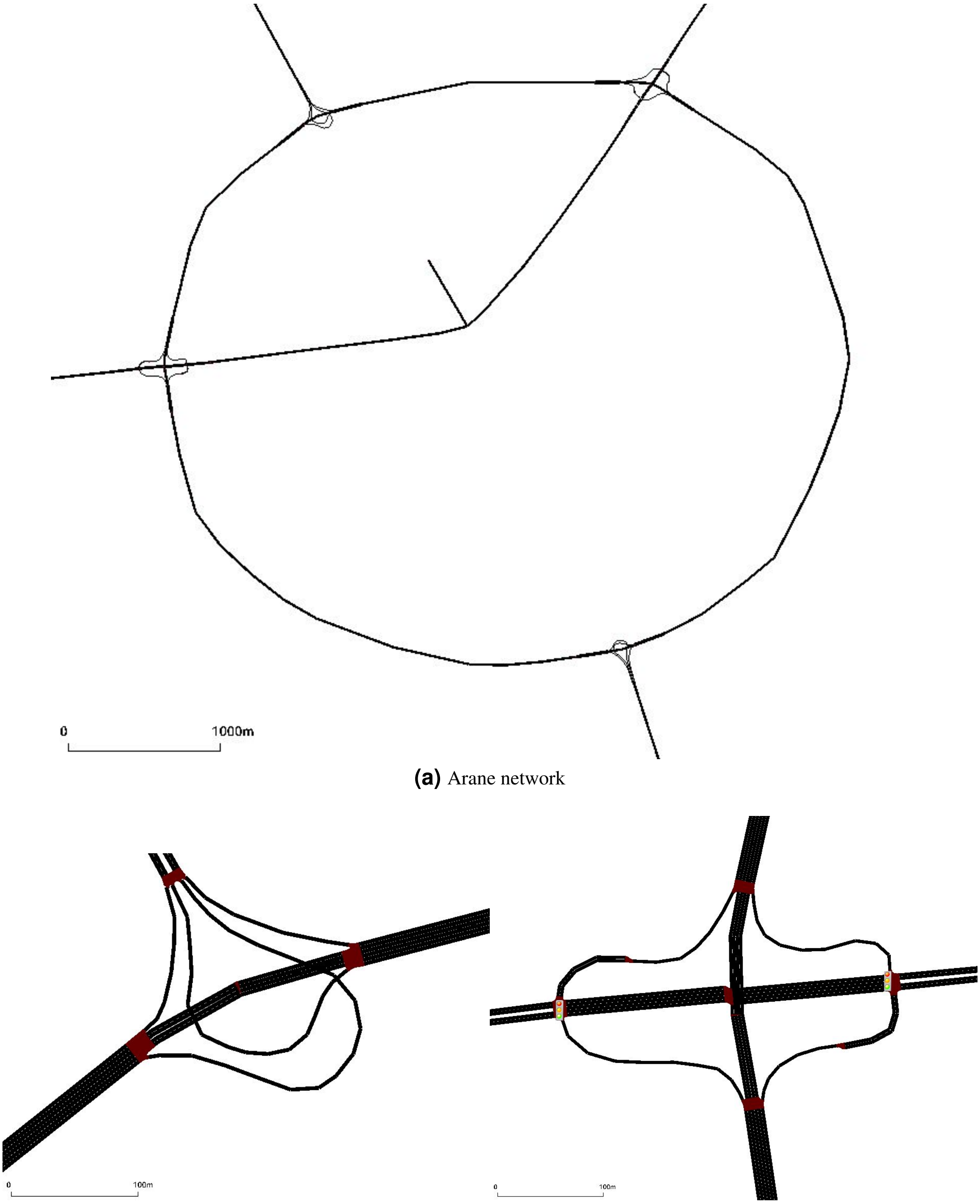
(a) Pi network



(b) Close-up intersection

Figure 6-1: Pi network and some close-up. [Own work]

When the future work will focus on larger networks some new challenges will arise. One of the main challenges will be the growing complexity of the optimisation problem, with more state variables, control variables and constraints. This will also induce re-tuning of the settings of the MPC controller for its control horizons and prediction horizons. For the prediction horizons, one needs to look again at the minimal travel time one must capture from point of control to the end of the network. The control horizon presumably needs to be kept small again.



(a) Arane network

(b) Close-up interchange

(c) Close up junction

Figure 6-2: Arane network and some close ups. [Own work]

One of the challenges for future work on a larger network, like the one given in figure 6-2, is to configure and build the network. The complexity of the configuration has been experienced during this thesis and should not be underestimated. Building intersections and junctions, especially with the interaction between city and freeway traffic, is time-consuming. The configuration itself has a certain complexity as well. Next to that, there are a lot of different demand settings one must set.

Furthermore, the speed of the optimisation algorithm is a real concern. For this reason, the type of algorithm, plus the type of code language needs to be reconsidered. With a larger network, the complexity of the optimisation problem increases. Two aspects are the main contributors to this. First, the number of services therefore, a longer prediction horizon and potentially a longer control horizon are needed. The second is due to longer travel times. Cars travel longer distances (with lower speeds) from point of control to their destination. This rule of thumb for the prediction horizon was also shared in section 3-4-2. For sake of estimation the travel time for the east and west ring of Amsterdam is taken as reference, see figure 6-3.

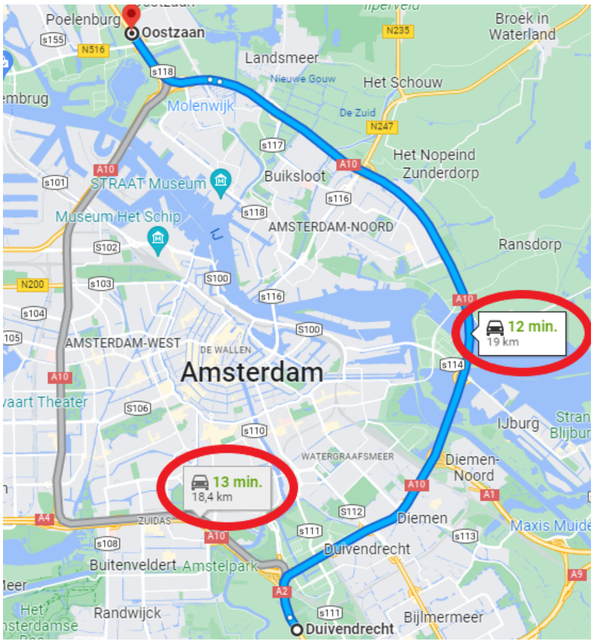


Figure 6-3: Travel time for the east and west ring of Amsterdam.

The average of the two would lead to a prediction horizon of 75 steps with $T_{\text{step}} = 10s$. Let's assume 20 different services/controls are required and a control horizon of 6 will do as well. This means a solution space of $2^{6*20} = 2^{120}$. A genetic algorithm in a python environment as proposed in this thesis would not fulfil to solve this optimisation problem. At this point, the MPC simulation for network 2 runs at a real-time factor of between 1.8 to 2. Based on simulations with the following processor: Intel(R) Core(TM) i5-9500 CPU @ 3.00GHz, 3000 Mhz, 6 Core(s), 6 Logical Processor(s). Let's assume that without SUMO time and only taking TraCI time into account, the GA solves the problem in $10s/2 = 5s$. If the GA would need to solve the new problem, one needs to increase population size (number of initial conditions) and the maximum number of iterations. Moreover, the limitation on *max iteration without improvement* should be removed or increased. The exact influence on the computation time is hard to predict and should be researched. When the target is a real-life implementation, a company could increase the budget and scale up the computational power. It could for instance increase the number of CPUs used on a single core from 1 to 4 and increase the number of CPUs between 4 to 8 leading to a maximum of 32 cores for a single PC. Increasing the number of PCs to 10 leads to a total of 320 cores. Furthermore, knowing that python is a relatively slow programming language, speed could be increased by a factor of 3 by switching to Rust or C. In total, the computation could be done around $1000 \approx 2^{10}$ times faster. The numbers presented in the previous two paragraphs are summarised in table 6-1.

Table 6-1: Numbers used for estimation on chance of success for the GA for a larger network.

Network	# services	Horizon	Solution space	GA time	PC upgrade
Simple freeway plus	2	6	2^{12}	$5s$	x1000 faster
Large network	20	6	2^{120}	$\dots$	x1000 faster

It is difficult to give a number on the exact computation time of the GA approach with pc upgrades. It is however the question if the smart features of the GA could cope with the $\approx 2^{100}$ increase in solution space, even when the PC upgrades provide a 1000 times faster computation. This should be tested. Nevertheless, it is preferred and advised to investigate a different optimisation strategy as well. Another possibility is to use a simplification of the model to use a different optimisation strategy. If one could rewrite the optimisation problem from a mixed integer nonlinear programming (MINLP) into a mixed integer linear programming (MILP), the optimisation algorithm could switch from a heuristic approach like the GA to a convex method. The simplification could be done by using a PWA approximation like done in [43]. An affine function is the composition of a linear function with a translation. The piecewise implies it will consist of multiple sub-function. A MILP can be solved by the Gurobi solver, which is a fast solver for these kinds of problems. The company itself claims that over the last 10 years they increased MIP solver performance by a factor of 53X. Exact improvements in computation time are hard to tell because it is compared to MIP solvers of 10 years ago. However, it is promising and worth researching.

Bibliography

[1] CREON, "Traffic centre of rijkswaterstaat." https://www.creon.nl/projecten/ministerie-van-rijkswaterstaat.html. (accessed: 17-05-2021).

[2] Rijkswaterstaat, "Het omleidingskaartje van a10 noord." http://www.amstelveenweb.com/nieuws-Snelweg-A10-Noord-wordt-afgesloten-van-31-juli-t&newsid=356092848. (accessed:17-05-2021).

[3] A. Hegyi, B. De Schutter, and H. Hellendoorn, "Model predictive control for optimal coordination of ramp metering and variable speed limits," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 3, pp. 185–209, 2005.

[4] S. Grammatico, *Model Predictive Control*. Lecture slides SC42125 TU Delft, 2018.

[5] K. CROW, *Handboek Verkeersmanagement Module Regelaanpak*. Crow Media, 2017.

[6] H. A. D. (AD), "Nederlandse wegen en ov naderen grenzen capaciteit." https://www.ad.nl/binnenland/nederlandse-wegen-en-ov-naderen-grenzen-capaciteit~a589d5ce/. (accessed: 18-03-2021).

[7] E. van der Aa, "Drukte op wegen en spoor gaan komende jaren toenemen." https://www.parool.nl/nieuws/drukte-op-wegen-en-spoor-gaan-komende-jaren-toenemen~ba58ba60/?referrer=https%3A%2F%2Fwww.google.com%2F. (accessed: 18-03-2021).

[8] M. van den Berg, B. D. Schutter, A. Hegyi, and J. Hellendoorn, "Model Predictive Control for Mixed Urban and Freeway Networks," *Proceedings of the 83rd Annual Meeting of the Transportation Research Board*, vol. 19, pp. 1–19, 2004.

[9] M. Van Den Berg, A. Hegyi, B. De Schutter, and H. Hellendoorn, "Integrated traffic control for mixed urban and freeway networks: A model predictive control approach," *European journal of transport and infrastructure research*, vol. 7, no. 3, pp. 223–250, 2007.

[10] J. Ault and G. Sharon, "Reinforcement Learning Benchmarks for Traffic Signal Control," no. NeurIPS, 2021.

[11] L. Heunks, "Literature survey on traffic network management; network-wide user-friendly and transparent traffic control for large-scale mixed city and freeway networks," 07 2021.

[12] G. Leduc, "Road Traffic Data : Collection Methods and Applications," *EUR Number: Technical Note: JRC 47967*, vol. JRC 47967, no. May, p. 55, 2008.

[13] SWOV, "Verkeersveiligheidscijfers." https://www.swov.nl/feitenencijfers/verkeersveiligheidscijfers. (accessed: 23-03-2021).

[14] RWS, "Actuele verkeersongevallencijfers." https://www.rijkswaterstaat.nl/wegen/wegbeheer/onderzoek/verkeersveiligheid-en-ongevallencijfers/actuele-verkeersongevallencijfers/index.aspx. (accessed: 23-03-2021).

[15] Nieuwsbericht, "Bijna 7,8 miljoen geconstateerde verkeersovertredingen in 2020." https://www.rijksoverheid.nl/actueel/nieuws/2021/02/10/bijna-78-miljoen-geconstateerde-verkeersovertredingen-in-2020#:~:text=Het%20aantal%20verkeersboetes%20na%20een,2019%20naar%20556.304%20in%202020., March 2021. (accessed: 23-03-2021).

[16] A. Messmer and M. Papageorgiou, "METANET: a macroscopic simulation program for motorway networks," *Traffic Engineering Control*, vol. 31, no. 8-9, pp. 466–470, 1990.

[17] C. F. Daganzo, "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transportation Research Part B*, vol. 28, no. 4, pp. 269–287, 1994.

[18] K. Aboudolas, M. Papageorgiou, and E. Kosmatopoulos, "Store-and-forward based methods for the signal control problem in large-scale congested urban road networks," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 2, pp. 163–174, 2009.

[19] S. Lin and Y. Xi, "An efficient model for urban traffic network control," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 17, no. 1 PART 1, pp. 14066–14071, 2008.

[20] S. Lin, B. De Schutter, Y. Xi, and J. Hellendoorn, "A simplified macroscopic urban traffic network model for model-based predictive control," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 42, no. 15, pp. 286–291, 2009.

[21] C. Pasquale, S. Sacone, S. Siri, and A. Ferrara, "Traffic control for freeway networks with sustainability-related objectives: Review and future challenges," jan 2019.

[22] F. Kessels, R. Kessels, and Rauscher, *Traffic flow modelling*. 2019.

[23] A. Kotsialos, M. Papageorgiou, C. Diakaki, Y. Pavlis, and F. Middelham, "Traffic Flow Modeling of Large-Scale Motorway Networks Using the Macroscopic Modeling Tool METANET," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 4, pp. 282–292, 2002.

[24] J. R. D. Frejo and B. De Schutter, "SPERT: A Speed Limit Strategy for Recurrent Traffic Jams," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 692–703, 2019.

[25] T. Toledo, H. Koutsopoulos, and M. Ben-Akiva, "Microscopic traffic simulation: Models and application," vol. 31, pp. 99–130, 01 2005.

[26] M. Saidallah, A. El Fergougui, and A. E. Elalaoui, "A comparative study of urban road traffic simulators," *MATEC Web of Conferences*, vol. 81, 2016.

[27] F. de Souza, O. Verbas, and J. Auld, "Mesoscopic traffic flow model for agent-based simulation," *Procedia Computer Science*, vol. 151, no. 2018, pp. 858–863, 2019.

[28] S. Calvert, "TrafficQuest report Traffic assignment and simulation models," 2016.

[29] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*, IEEE, 2018.

[30] J. Barcelo, *Fundamentals of Traffic Simulation*, vol. 145 of *International Series in Operations Research Management Science*. New York, NY: Springer New York, 2010.

[31] D. Krajzewicz, B. Heldt, S. Nieland, R. Cyganski, and K. Gade, "Sustainable urban mobility and communting in Baltic cities: Guidance for transport modelling and data collection," *Baltic Sea Region Mobility Summit*, 2019.

[32] C. Pasquale, S. Sacone, S. Siri, and B. De Schutter, "A multi-class model-based control scheme for reducing congestion and emissions in freeway networks by combining ramp metering and route guidance," *Transportation Research Part C: Emerging Technologies*, vol. 80, pp. 384–408, 2017.

[33] J. R. D. Frejo and B. De Schutter, "Feed-Forward ALINEA: A Ramp Metering Control Algorithm for Nearby and Distant Bottlenecks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 7, pp. 2448–2458, 2019.

[34] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn, "Efficient network-wide model-based predictive control for urban traffic networks," *Transportation Research Part C: Emerging Technologies*, vol. 24, pp. 122–140, 2012.

[35] J. R. D. Frejo, A. Núñez, B. De Schutter, and E. F. Camacho, "Hybrid model predictive control for freeway traffic using discrete speed limit signals," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 309–325, 2014.

[36] B. L. Ye, W. Wu, K. Ruan, L. Li, T. Chen, H. Gao, and Y. Chen, "A survey of model predictive control methods for traffic signal control," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 3, pp. 623–640, 2019.

[37] N. Groot, B. De Schutter, and H. Hellendoorn, "Toward system-optimal routing in traffic networks: A reverse stackelberg game approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 29–40, 2015.

[38] E. Walraven, M. T. Spaan, and B. Bakker, "Traffic flow optimization: A reinforcement learning approach," *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 203–212, 2016.

[39] C. Pasquale, S. Sacone, S. Siri, and A. Ferrara, "Supervisory multi-class event-triggered control for congestion and emissions reduction in freeways," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2018-March, no. October, pp. 1–6, 2018.

[40] C. Pasquale, S. Sacone, S. Siri, and A. Ferrara, "A multi-class decentralised event-triggered control framework for congestion and emission reduction in freeway networks," *IFAC-PapersOnLine*, vol. 51, no. 9, pp. 291–298, 2018.

[41] J. Rawlings and D. Mayne, *Model Predictive Control: Theory Design.* Nob Hill Pub, 2009.

[42] A. Kotsialos, M. Papageorgiou, and A. Messmer, "Optimal coordinated and integrated motorway network traffic control," in *14th International Symposium on Transportation and Traffic TheoryTransportation Research Institute*, 1999.

[43] N. Groot, B. D. Schutter, S. Zegeye, and H. Hellendoorn, "Model-based predictive traffic control: A piecewise-affine approach based on METANET," vol. 19, 2011.

[44] R. M. Solgi, "geneticalgorithm 1.0.2." https://pypi.org/project/geneticalgorithm/#1112-id. (accessed:24-02-2022).

[45] M. Van den Berg, A. Hegyi, B. De Schutter, and J. Hellendoorn, "A Macroscopic Traffic Flow Model for Integrated Control of Freeway and Urban Traffic Networks," *Proceedings of the IEEE Conference on Decision and Control*, vol. 3, no. December, pp. 2774–2779, 2003.

[46] H. R. Kashani and G. N. Saridis, "Intelligent control for urban traffic systems," *Automatica*, vol. 19, no. 2, pp. 191–197, 1983.

[47] N. Wu, D. Li, Y. Xi, and B. de Schutter, "Distributed Event-Triggered Model Predictive Control for Urban Traffic Lights," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2020.

[48] G. A. C. D. SUMO eclipse *et al.*, "Roadcapacity sumo." `https://sumo.dlr.de/docs/Simulation/RoadCapacity.html#further_headway_effects`. (accessed:01-03-2022).

[49] J. R. D. Frejo, E. Camacho, and R. Horowitz, "A parameter identification algorithm for the metanet model with a limited number of loop detectors," pp. 6983–6988, 12 2012.

Glossary

List of Acronyms

CTM	Cell Transmission Model
FCD	Floating Car Data
GA	Genetic Algorithm
LoS	Level of Service
LRA	"Landelijke Regelaanpak"
MILP	mixed integer linear programming
MINLP	mixed integer nonlinear programming
MPC	Model Predictive Control
NM	Network Manager
PWA	Piecewise Affine
SFM	Store and Forward Model
TTS	Total Time Spent
VMS	Variable Message Signs