Model Predictive Control for Large Freeway Networks
An approach using a distributed control architecture and alternating optimisation

Uglješa Todorović
Model Predictive Control for Large Freeway Networks
An approach using a distributed control architecture and alternating optimisation

MASTER OF SCIENCE THESIS

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

Uglješa Todorović

November 14, 2019

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology
Due to the yearly increase in the number of vehicles and the need for transportation, traffic congestion has become a crucial problem in today’s society. There is a need for a sustainable solution to reduce or even eliminate traffic jams. Freeway traffic control has shown to be a sustainable solution to this problem. Especially the implementation of Ramp Metering (RM) installations and Variable Speed Limits (VSLs) as control measures is currently a widely researched area because the proper coordination of those measures can significantly reduce traffic congestion, traffic emissions and the risk of accidents.

The Model Predictive Control (MPC) framework has shown outstanding capabilities for coordinated control of RM installations and VSLs. However, the inherent nonlinearity of traffic flow, in combination with the discrete nature of VSLs and the continuous nature of RM rates, yields a Mixed Integer Nonlinear Programming (MINLP) optimisation problem that has to be solved within every controller sampling interval. The computational time that is needed to solve MINLP problems generally increases exponentially with the size of the problem. Therefore, the implementation of MPC to large freeway networks remains challenging.

This work proposes two novel MPC algorithms for coordinated control of continuous RM rates and discrete VSLs on large freeway networks. Both algorithms use a distributed control architecture and an alternating optimisation scheme to relax the MINLP problems and, hence, offer a trade-off between computational complexity and system performance. A case study is performed to evaluate the performance of both algorithms. In this case study a 30 km long freeway network is used that contains six VSLs and three RM installations.

The first part of the case study shows that relaxing the VSLs to be continuous decision variables instead of discrete decision variables in the optimisation problems results in a major performance loss with a distributed architecture. This result contrasts with many related works, where the MINLP optimisation problems are relaxed by considering the VSLs to be continuous decision variables.

The second part of the case study evaluates the proposed distributed algorithms by comparing their performance to the more conventional centralised and decentralised MPC algorithms. Both proposed algorithms have a lower computational complexity than the centralised algorithm, as they manage to solve the optimisation problems within the controller sampling intervals. Moreover, one of the proposed algorithms has a system performance that is remarkably similar to the optimal performance of the centralised algorithm.
# Table of Contents

Acknowledgements vii

1 Introduction 1
  1-1 Motivation .......................................................... 1
  1-2 Freeway traffic control ............................................. 2
    1-2-1 Aim of freeway traffic control ............................... 2
    1-2-2 Coordination of control measures: MPC approach ........... 3
    1-2-3 Difficulty: computational complexity ......................... 4
  1-3 Problem statement .................................................. 5
  1-4 Thesis outline ...................................................... 5

2 Literature Review: MPC for Large Freeway Networks 7
  2-1 Modelling of freeway traffic ....................................... 7
    2-1-1 Types of traffic models ....................................... 7
    2-1-2 Macroscopic traffic models ................................... 8
    2-1-3 Conclusions ..................................................... 9
  2-2 MPC objective functions ........................................... 9
    2-2-1 Congestion reduction .......................................... 9
    2-2-2 Penalty on variation in ramp metering rates ............... 10
    2-2-3 Soft constraints on queue lengths ............................ 10
    2-2-4 Conclusions .................................................... 10
  2-3 MPC Architectures .................................................. 10
    2-3-1 Centralised MPC ............................................... 11
    2-3-2 Decentralised MPC ............................................. 11
    2-3-3 Distributed MPC ............................................... 11
    2-3-4 Performance comparison ....................................... 13
    2-3-5 Conclusions .................................................... 13
# Table of Contents

2-4 Mixed integer optimisation
- 2-4-1 Approaches ................................. 14
- 2-4-2 Discrete optimisation ..................... 15
- 2-4-3 Nonlinear non-convex continuous optimisation .......... 15
- 2-4-4 Conclusions ............................... 16

2-5 Conclusions ..................................... 16

3 Distributed MPC for Large Freeway Networks using Alternating Optimisation ........... 17
- 3-1 Freeway network .............................. 17
- 3-2 Control algorithms ......................... 18
  - 3-2-1 Overview .................................. 18
  - 3-2-2 Distributed architecture ................ 19
  - 3-2-3 Operational constraints ................. 21
  - 3-2-4 Alternating optimisation ................. 21
  - 3-2-5 Initialisation and stopping criteria ....... 25
  - 3-2-6 Algorithm formulation .................... 26
- 3-3 Conclusions ................................. 27

4 Case Study and Comparison ...................... 29
- 4-1 Case study details ............................. 29
  - 4-1-1 Freeway network and simulation .......... 29
  - 4-1-2 Performance criteria ..................... 31
  - 4-1-3 Computer specifications .................. 32
  - 4-1-4 No-control system response ............... 32
- 4-2 Control parameters ........................... 33
  - 4-2-1 System partitioning ...................... 33
  - 4-2-2 Sampling time and horizons ............... 34
  - 4-2-3 Weights of objective functions and constraints .. 35
  - 4-2-4 Number of iterations ..................... 35
- 4-3 Case study A: Relaxation on VSLs ............. 37
  - 4-3-1 Results .................................. 38
  - 4-3-2 Conclusions ............................. 40
- 4-4 Case study B: Architecture comparison ............ 40
  - 4-4-1 Results .................................. 40
  - 4-4-2 Comparison .............................. 48
  - 4-4-3 Conclusions .............................. 51
- 4-5 Conclusions ................................. 51

5 Concluding Remarks ............................... 53
- 5-1 Conclusions .................................. 53
  - 5-1-1 Research questions ....................... 53
  - 5-1-2 Contributions ............................ 55
- 5-2 Future work .................................. 56
Acknowledgements

First of all, I would like to thank my daily supervisor dr. José Ramón Domínguez Frejo for his continuous support during my research. I could not have wished for a better daily supervisor. His insights and enthusiasm, and the numerous discussions that we had, were crucial to the final outcome of this Master thesis.

I would also like to thank my supervisor prof.dr.ir. Bart De Schutter for his guidance during my thesis. This thesis would definitely not be of the same quality without his professional attitude, critical questions and remarks.

I owe much gratitude towards my parents, sister and grandma: Milovan, Smiljka, Teodora and Milka. No matter with what (seemingly impossible) challenge or obstacle I have been faced during my thesis, but also during my life in general, they have always known to motivate me with their unconditional love and support. Hvala vam što ste uvijek tu u mom životu.

Without a doubt, I need to thank my study colleagues with whom I enjoyed countless lunch and coffee breaks. You are a great part of the reason why I enjoyed my master studies so much.

Lastly, I cannot end my acknowledgements without mentioning de Apenhots, Gravitas, Eco-Runner Team VII and all the other amazing friends I have made during my studies. You have made my experience in Delft unforgettable. Thank you.

Delft, University of Technology
November 14, 2019

Uggi Todorović
“All that was great in the past was ridiculed, condemned, combated, suppressed – only to emerge all the more powerfully, all the more triumphantly from the struggle.”

— Nikola Tesla
Chapter 1

Introduction

1-1 Motivation

As the number of vehicles and the need for transportation keeps increasing every year, traffic congestion has become a crucial problem in today’s society. Especially in densely populated areas during rush hours, capacities of freeways are exceeded daily, resulting in traffic jams. The Kennisinstituut voor Mobiliteit\(^1\) has predicted that travel times of motorists in the Netherlands will increase with 35\% by the year 2023, mainly due to congestion of freeways\([32]\).

This increase in traffic congestion has various negative consequences for society. Firstly, it has a negative impact on human health, since the probability of traffic accidents and mental stress increases. Furthermore, there is a cost associated with the time wasted by people standing in traffic jams. Lastly, traffic congestion has a negative impact on the environment because it results in higher emissions and waste of fuel.

The negative impact on the environment is a growing concern. Climate change is one of the toughest challenges humanity has ever faced. Because of this, the Paris Agreement\([51]\) is signed within the United Nations Framework Convention on Climate Change to keep the increase in the global average temperature of the earth to well below 2 °C above pre-industrial levels.

CO\(_2\) emissions are a major contributor to global warming. In the Netherlands in 2017, 16\% of the overall CO\(_2\) emissions were caused by road traffic\([13]\) and a large part of these traffic emissions come from traffic jams on freeways\([9]\). Ideally, we want a society where all cars have net-zero emissions, yet this is an exceptionally challenging task. Before we reach the point where all vehicles can have net-zero emissions, an intermediate solution could be to at least reduce the number of freeway traffic jams.

Various approaches can be taken to reduce the number of traffic jams. One obvious way could be to build additional freeway infrastructure. However, this is only a temporary solution, since

\(^1\)Kennisinstituut voor Mobiliteit is the Dutch Institute for Transport Policy Analysis.
the number of vehicles that use freeways keeps increasing. Additionally, the construction of freeways has a negative impact on the environment due to the destruction of nature and is very costly.

Another approach that has gained increasing attention over the past few years is the implementation of congestion tax. This is a taxing system where motorists are obliged to pay a tax when entering freeways during rush hours to reduce the number of vehicles on the freeways. Although this system has already been successfully implemented in Stockholm [11], this idea has been rejected in the Netherlands. The main reasons for the rejection were that it had been predicted that urban roads would get overloaded because motorists would start taking indirect routes [21], the implementation of such a system is very costly, and it is questionable if congestion tax would even be implementable in a country with such a vast amount of traffic like the Netherlands [21, 54].

A currently widely researched approach is freeway traffic control, which uses real-time measurements of the traffic state to determine inputs for control measures in order to reduce traffic congestion or achieve other traffic objectives. Various studies have shown that freeway traffic control can be a sustainable solution to the increasing demand of vehicles on freeways and the associated increase in traffic congestion [4, 6, 44, 50]. Hence, this work will use freeway traffic control for the reduction of traffic congestion on freeways.

The control measures that are used with freeway traffic control have to be coordinated at a large spatial scale to achieve optimal performance, since local control actions have significant influences on the traffic state of distant parts of the freeway network [26]. Although much research has been done in such advanced controllers that coordinate control measures at a large scale, currently there are only relatively simplistic, local controllers implemented in practice. One of the main reasons why such advanced controllers are hard to implement is because they are usually computationally very complex. This thesis aims to design a control method that coordinates control measures of freeways at a large spatial scale, and therewith offers a good trade-off between computational complexity and system performance.

1-2 Freeway traffic control

1-2-1 Aim of freeway traffic control

The goal of traffic controllers can vary greatly depending on the desires of the stakeholders of a freeway network, time of operations of a network and location of a network [42]. While environmentalist might want to reduce the dispersion of emissions of a freeway network, transport authorities might prefer to increase the network throughput and safety. In this thesis, the objective of traffic control is to reduce traffic congestion and therefore increase traffic throughput. With the reduction in traffic congestion, indirectly the overall vehicular emissions are reduced and traffic safety is increased [53].

To illustrate how a reduction in traffic congestion can be achieved, some basic freeway traffic modelling concepts have to be introduced. A common way in literature to describe the dynamics of freeway traffic is to describe traffic flow from the viewpoint of collective vehicular flow, i.e. from a macroscopic viewpoint. Generally, the state of a freeway section is then described by three aggregated variables: the vehicular density $\rho$ (in veh/km/lane), the vehicular flow $q$ (in veh/h) and the vehicular space-mean speed $v$ (in km/h).
The steady-state relationship between the three aggregated variables is usually described by a Fundamental Diagram (FD) [3]. Although the shape of the diagram differs per model, a typical FD is depicted in Figure 1-1.

![Flow-density Curve](image1)

**Figure 1-1:** An example of a typical FD (described by METANET [35] with the parameter values from [24]), describing the steady-state relationship between desired speed, density and flow.

In uncongested traffic state, the traffic flow increases as the traffic density increases, i.e. as more vehicles enter the freeway. However, as can be seen in Figure 1-1, there is a critical point, where suddenly the traffic flow decreases as more vehicles enter the freeway. At this critical point, described by the critical density, capacity flow and critical speed, congestion starts to occur. Hence, freeway traffic control aims to keep the densities of all freeway sections below this critical density, so that all the freeway sections operate in an uncongested state.

### 1-2-2 Coordination of control measures: MPC approach

The types of traffic control measures that are available for control engineers varies greatly per country. However, Variable Speed Limits (VSLs) and Ramp Metering (RM) installations are currently two of the most implemented and researched control measures. RM installations control the number of vehicles that enter freeways from on-ramps. This is usually done with the implementation of a traffic light at the on-ramp. The entering vehicle flow is then controlled by adjusting the green phase lengths of the traffic signal.

VSLs are flexible restrictions on the speed at which motorists are allowed to drive on a stretch of freeway, usually displayed on variable message signs. These speed limits are only allowed to take values from a prescribed discrete set, which varies per country. Implementations of both control measures in the Netherlands are illustrated in Figure 1-2.

Various studies have shown that with the proper coordination of VSLs and RM installations, a balanced delay-time between vehicles on the mainline and vehicles waiting at on-ramps can be achieved, while spillback from on-ramp queues to urban roads can be avoided [55]. However, it is crucial for good performance that those control measures are coordinated at a
large spatial scale, since local control actions have significant influences on the traffic state of distant parts of the freeway network [26].

The Model Predictive Control (MPC) framework [17] has shown outstanding capabilities for coordinated control of VSLs and RM installations [5, 8, 20, 23, 25]. MPC is an advanced control methodology, where an objective function is minimised to find optimal control inputs, and a prediction model is used to predict relevant future system trajectories. With the optimisation, multiple control inputs can be determined, and the prediction model can be used to predict influences of those control inputs on distant parts of the network. Therefore, the MPC framework is highly suitable for the coordination of VSLs and RM installations. Additionally, constraints can be imposed on the states and inputs in the optimisation. Consequently, maximum queue lengths at on-ramps can be imposed, so that spill-back to urban roads can be avoided.

For those reasons, this work will use the MPC framework for coordinated control of VSLs and RM installations on large freeway networks. For an extensive description and overview of these and other control measures, the reader is referred to [34].

1-2-3 Difficulty: computational complexity

Related to the scope of this work, there are three aspects in particular that make the implementation of MPC to large freeway networks challenging:

- The dynamics of freeway traffic flow are inherently nonlinear. Therefore, generally, nonlinear prediction models are used in literature (e.g. in: [20, 25, 44]).
- The VSLs are, by law, only permitted to take values from a discrete set, while the RM rates are continuous control inputs.
- MPC has to be applied at a large spatial scale for good performance, so that multiple control control measures are coordinated.

The combination of a nonlinear prediction model with discrete VSLs and continuous RM rates yields a Mixed Integer Nonlinear Programming (MINLP) optimisation problem that has to
be solved within every controller sampling interval. Generally, the computational time that is needed to solve such a problem scales exponentially with the size of the problem. Hence, the computational time quickly becomes larger than the controller sampling time with the application of MPC to large freeway networks, making it unimplementable.

A distributed control architecture [16] is a popular approach that is used to reduce the computational complexity of large-scale MPC problems in general. However, the implementation of such an architecture to freeway networks is difficult, as every distributed agent still has to solve an MINLP problem. Hence, the existing distributed approaches either do not consider VSLs as control inputs [1, 2, 19], or relax these to be continuous decision variables in the optimisation problems [25, 26].

1-3 Problem statement

The goal of this thesis is to answer the following main question:

*Is it possible to develop distributed MPC algorithms for large freeway networks using discrete variable speed limits and continuous ramp metering rates that offer a good trade-off between computational complexity and system performance?*

The criteria that are used for system performance and computational complexity in this work are defined in Section 4-1-2. Two sub-questions are derived to answer the main question:

1. *Does relaxing the variable speed limits to be continuous decision variables in the optimisation result in a significant performance loss with a distributed architecture?*

   Most MPC approaches that use VSLs and RM installations greatly simplify the optimisation problems by relaxing the VSLs to be continuous decision variables. However, related works have contradictory conclusions on this relaxation, as some have found that it results in a large performance loss [23]. Hence, the effect of this relaxation on the system performance is still an open question in the field. Therefore, the effects of this relaxation with a distributed control architecture are investigated in Section 4-3.

2. *How is the performance of the proposed distributed algorithms compared to centralised and decentralised algorithms in terms of system performance and computational complexity?*

   In general, centralised MPC has the optimal system performance in a receding horizon context, but is also computationally very complex. Vice versa, decentralised MPC generally has the worst system performance, but is also significantly less complex from a computational viewpoint. The performance of both architectures is used as a reference to evaluate the performance of the developed distributed algorithms in Section 4-4.

1-4 Thesis outline

The rest of this thesis is structured as follows. Chapter 2 presents relevant literature on the application of MPC to large freeway networks. Chapter 3 proposes two novel distributed MPC algorithms for coordinated control of discrete VSLs and continuous RM
rates for large freeway networks, called Fully Cooperative Alternating Model Predictive Control (FC-A-MPC) and Downstream Cooperative Alternating Model Predictive Control (DC-A-MPC). Chapter 4 presents a case study that evaluates the performance of the proposed distributed algorithms. Finally, Chapter 5 concludes this thesis and suggests areas for future work.
Chapter 2

Literature Review: MPC for Large Freeway Networks

This chapter outlines the related work on the application of MPC to large freeway networks using continuous RM rates and discrete VSLs. The reader is expected to be familiar with the basic concepts of MPC. In Appendix A, these concepts are summarised briefly and if further reading is necessary, the reader is referred to [17]. This chapter is structured as follows. Section 2-1 outlines the different approaches for freeway traffic modelling. Then, Section 2-2 discusses objective functions that can be used with the application of MPC to freeway control. Subsequently, Section 2-3 outlines three different control architectures for the application of MPC. Section 2-4 discusses approaches for solving MINLP optimisation problems, which are the type of optimisation problems that have to be solved in this work. Lastly, the chapter is concluded in Section 2-5.

2-1 Modelling of freeway traffic

The MPC framework uses a prediction model of the system to obtain relevant future system trajectories. This section first gives a general overview of the different modelling approaches that have been developed for freeway traffic flow. Then, different macroscopic models are outlined, since these are found to be the most suitable as prediction models. Finally, a conclusion is made on the prediction model that will be used in this work.

2-1-1 Types of traffic models

Most of the current traffic models can be categorised by three levels of detail [47]:

- **Microscopic traffic models.** Microscopic traffic models describe vehicles individually with full information on vehicle trajectories. They make assumptions on the individual
behaviour of drivers. Therefore, these models incorporate a high-detailed description of the motion and interactions of each vehicle. Microscopic models are commonly used for detailed simulation because they allow studying of individual vehicle motion. An extensive comparison between different microscopic models is made in [52].

- **Macroscopic traffic models.** Generally, macroscopic models describe traffic flow from the viewpoint of collective vehicular flow. Hence, a low-detailed description of the traffic flow is given, as individual vehicle motions and interactions are neglected. Often, aggregated variables such as density, mean speed and flow based on hydrodynamic analogies are used. Therefore, the computational demand of the models generally does not increase when the traffic density increases.

- **Mesoscopic traffic models.** Mesoscopic models describe the behaviour of vehicles in probabilistic terms, without explicitly distinguishing their space-time behaviour. They represent a medium-detailed description of traffic flow. Examples of mesoscopic models are headway distribution models [15] and gas-kinetic models [45].

Mesoscopic and microscopic models are not suitable for real-time model-based control because they have a relatively high computational complexity. On the contrary, many macroscopic models can capture enough dynamics for real-time traffic control [30], while their computational complexity is relatively low. Hence, the majority of related approaches use macroscopic traffic models (e.g. in: [1, 4, 25]) and, therefore, the rest of this work uses a macroscopic modelling approach as well. The following section gives an overview of macroscopic traffic modelling. For an overview of microscopic and mesoscopic traffic models, the reader is referred to [47].

### 2-1-2 Macroscopic traffic models

The accuracy of macroscopic models usually increases as the model complexity increases, but so does the computational complexity. Generally, only higher-order macroscopic models are capable of predicting complex traffic phenomena like stop-and-go waves at bottlenecks and capacity drops. Hence, a balanced trade-off between model complexity and computational complexity is necessary for good performance. The majority of related works only use two macroscopic models and extensions thereof: METANET [35] and the Cell Transmission Model (CTM) [14].

CTM is a first-order model, where a static speed-density relationship describes the traffic flow dynamics. METANET is a second-order model, where the speed dynamics are described in a heuristic way. Due to the higher model complexity, METANET simulates congested traffic with significantly more accuracy than CTM [22]. Both CTM and METANET are nonlinear models. These nonlinearities inherently result in high computational complexities. Both models can be rewritten as Mixed Logical Dynamical (MLD) models to get rid of the nonlinearities [1, 36].

For a more detailed overview of these and other macroscopic traffic models, the reader is referred to [47].
2-2 MPC objective functions

2-1-3 Conclusions

The macroscopic traffic model METANET will be used as the prediction model, as it has proven a good trade-off between model accuracy and computational complexity [5, 6, 25]. However, all the proposed methods are independent of the prediction model and therefore can equivalently be applied to other macroscopic models. The system dynamics of METANET are described in Appendix B.

2-2 MPC objective functions

An objective function is optimised to find optimal control inputs. This objective function represents the goal that the MPC wants to achieve. If more than one goal needs to be achieved, the overall objective function can be obtained by addition of the sub-objectives with appropriate weights [17]:

$$J(k_c) = \sum_{i=1}^{N_{\text{obj}}} \zeta_i J_i(k_c), \quad (2-1)$$

where $J(k_c)$ is the comprehensive objective function at controller sample $k_c$, $\zeta_i$ is the weighting factor for sub-objective $J_i(k_c)$ and $N_{\text{obj}}$ is the number of sub-objectives. This work considers a term for the reduction of congestion, a term that penalises input signal fluctuation, and soft constraints in the comprehensive objective function, similarly to [24]. The rest of this section outlines these terms.

Remark. All objective functions are formalised with the notation and modelling concepts of METANET. If the reader is not familiar with those concepts, it is advisable to first read Appendix B. Moreover, only a division between segments is made with the modelling of networks to improve readability. This is possible because the network in the case study in Chapter 4 can be modelled by one large link.

2-2-1 Congestion reduction

The majority of papers in the field (e.g. in: [1, 2, 24, 25]) use the Total Time Spent (TTS) as the performance index for the reduction of congestion. The TTS is the summation of the time that all vehicles spend in a section, which at $k_c$ over the prediction horizon $N_p$ is defined as:

$$J_{\text{TTS}}(k_c) = \sum_{k=Mk_c}^{M(k_c+N_p)} T_c \left( \sum_{i \in I_{\text{all}}} \rho_i(k) L_i \lambda + \sum_{i \in I_t} w_i(k) \right), \quad (2-2)$$

where $T_c$ is the sampling time of the controller, $I_{\text{all}}$ is the set of all the considered segments $i$ in the objective function, $I_t$ is the set of all the considered segments containing a controlled RM installation, $M$ is a positive integer that relates the control time step $k_c$ and $k$ as $k = Mk_c$, $\rho_i(k)$ is the density of segment $i$, $\lambda$ is the number of lanes on the considered freeway section and $w_i(k)$ is the queue length at the on-ramp of segment $i$. 

Master of Science Thesis

Uglješa Todorović
2-2-2 Penalty on variation in ramp metering rates

It is common practice (e.g. in: [5, 24, 25, 50]) to include a term in the objective function that penalises the variation in ramp metering rates to avoid large and frequent fluctuations in the control signal:

\[
J_r(k_c) = \sum_{j=1}^{N_p-1} \left( \sum_{i \in I_r} \left( U^i_r(k_c + j) - U^i_r(k_c + j - 1) \right)^2 \right),
\]

where \( U^i_r(k_c) \in [0,1] \) is the metering rate at segment \( i \).

2-2-3 Soft constraints on queue lengths

Instead of imposing hard constraints on the queue lengths in the optimisation problem, it is suggested in [24] to use soft constraints in the objective function:

\[
J_w(k_c) = \sum_{k=M_{k_c}}^{M_{k_c+N_p}} \left( \sum_{i \in I_r} \max \left( w^i_i(k) - w_{\text{max}}, 0 \right)^2 \right).
\]

where \( w_{\text{max}} \) is the maximum imposed queue length. This can be beneficial, as sometimes a slight violation in queue lengths can result in an overall much better global system performance. Additionally, it simplifies the optimisation problems by reducing the number of hard constraints on the states of the system.

2-2-4 Conclusions

The objective functions in this work will consider three terms. The first term expresses the TTS for the reduction of traffic congestion. The second term penalises the variation in ramp metering rates to avoid large and frequent fluctuations in the control signal. The last term expresses a soft constraint on the queue lengths at the on-ramps.

2-3 MPC Architectures

Two conventional control architectures were used in the past for the application of MPC: centralised schemes, where one central agent determines inputs for the overall system, and decentralised schemes, where the system is partitioned into subsystems and each subsystem has a personal agent determining inputs based on local information. However, often for large-scale systems, centralised MPC is computationally too complex, and decentralised MPC lacks performance [1, 26]. The significant advancement of information technology and computer science [39] has driven the development of distributed architectures for the control of large-scale systems. With distributed MPC, control inputs can be coordinated for good performance, while the computational complexity can be reduced significantly [1, 19, 25].

In the remainder of this section, an overview is given of the application of the three architectures to freeway traffic control. Subsequently, their performance is compared. Finally, a conclusion is made on the distributed architectures that will be used in this work.
2-3 MPC Architectures

2-3-1 Centralised MPC

With centralised MPC, the freeway network is modelled by one comprehensive model and one central MPC agent determines inputs for the comprehensive network. The inputs are found by optimising one global objective function, using measurements of all states of the network. The consequence of minimising one global objective function is that the control inputs are optimally coordinated because all the system interactions are taken into account [17]. Therefore, centralised MPC generally leads to optimal system performance in a receding horizon context [25, 26].

Although centralised control provides the optimal solution in terms of system performance, it is generally not applicable to large-scale systems such as freeways networks. The main issue with centralised MPC is that the size of the optimisation problem grows with the size of the network. Since an MINLP has to be solved within every controller sampling interval, for which the computational time typically grows exponentially with the size of the problem, it quickly occurs that the optimisation problem cannot be solved within one controller sampling interval.

2-3-2 Decentralised MPC

With a decentralised MPC architecture, the freeway network is partitioned into subsystems and each subsystem is controlled by a local MPC agent. The agents determine local control inputs by optimising a local objective function, based on local measurements only. A generic decentralised MPC scheme is depicted in Figure 2-1.

![Figure 2-1: A generic decentralised MPC architecture for a freeway traffic system.](image)

By splitting the network into subsystems, the computational complexity does not scale with the size of the network, but instead with the size of the subsystems. However, the control inputs are not coordinated because the agents only use local information. Therefore, decentralised MPC can be an attractive architecture for systems where subsystems do not interact significantly. However, for freeway traffic control, this generally results in a significant performance loss [1, 19], as control inputs can even counteract the actions of each other [26].

2-3-3 Distributed MPC

An intermediate solution to the drawbacks of centralised and decentralised MPC is a distributed MPC architecture. With distributed MPC, the system is partitioned into subsystems
and all subsystems are controlled by local agents, similar to decentralised control. However, the agents are actively communicating or cooperating to improve global network performance. Commonly, the subsystems are defined as subsets of contiguous freeway portions which contain one or more control measures [3]. An alternative is to consider subsystems with only one control measure. Subsequently, the centralised problem can be reformulated into sets of single-input control problems [3]. A generic MPC architecture is illustrated in Figure 2-2.

![MPC architecture diagram](image)

**Figure 2-2:** A generic distributed MPC architecture for a freeway traffic system.

### Differences in distributed algorithms

Since distributed MPC is a generic term for all the decentralised MPC approaches where agents at least share some information whilst determining local inputs, there is a great variety in distributed algorithms. Hence, there is no single generally accepted approach [39]. Instead, much research has been done on the applicability of different types of distributed algorithms for different classes of systems. Surveys have been conducted on the categorisation and comparison of different distributed approaches [39, 41]. Relevant to freeway traffic control, distributed algorithms can differ in three key elements [39, 41]:

- **Iterative or non-iterative.** In iterative schemes, agents exchange information during the optimisation of the objective function. They find an intermediate solution to their optimisation problem, communicate the results to other agents, and resolve the optimisation problem with the updated values of other agents. By doing so, a convergence of the variables shared by multiple agents can be achieved [39].

  The number of iterations that is possible depends on the sampling time of the controller and the computational time required per iteration. However, the computational time required per iteration decreases rapidly because each optimisation is generally started in the optimum obtained in the previous iteration [25].

  The majority of distributed algorithms are designed for linear systems, for which often convergence by increasing the number for iterations can be proven based on the convexity properties of the problem [33]. As most freeway traffic control approaches use nonlinear prediction models, convergence is often investigated by means of simulations [26]. Other approaches linearise the prediction model [1, 36].

- **Cooperative or non-cooperative.** With cooperative algorithms, agents consider system-wide or neighbouring objectives in their optimisation, whereas in non-cooperative...
algorithms, objective functions only include local variables. Hence, agents in cooperative algorithms may sacrifice in terms of local performance to attain a better global performance [39]. It has been shown that non-cooperative algorithms converge to a Nash Equilibrium [49], which in general is not Pareto optimal [31]. Therefore, non-cooperative algorithms generally do not converge to the optimal performance of centralised MPC.

A variety of cooperative approaches for freeway traffic have been studied. Some use a fully cooperative approach [19, 25], where all agents share the same objective function. In other approaches, agents share their objective function with upstream [19], downstream [19] or both upstream and downstream agents [1, 19]. Generally, the performance increases as the cooperativeness increases. However, so does the computational complexity.

- **Serial or parallel.** In serial schemes, agents optimise their objective function one after another, whereas in parallel schemes, the agents optimise their objective function at the same time. Generally, serial schemes require fewer iterations to converge because agents make use of the most up-to-date information from their neighbours [40]. However, serial schemes are less scalable than parallel schemes [33].

### 2-3-4 Performance comparison

In [1, 2, 19, 25, 26], distributed and decentralised MPC schemes are compared to the optimal solution of centralised MPC for freeway traffic control. Decentralised MPC results in a significant performance loss compared to the optimal solution [1, 2, 19, 25]. A fully cooperative approach often leads to a similar system performance as centralised MPC [19, 25], but still has a significant computational load. The approaches where agents share their objective function with downstream agents have shown decent system performance and a significantly reduced computational complexity [19]. Additionally, the computational complexity does not scale with the size of the freeway network. However, both centralised and fully cooperative approaches have a better system performance.

### 2-3-5 Conclusions

The fully cooperative and the downstream cooperative architectures will be used for the distributed algorithms. Both architectures have shown a good trade-off between computational complexity and system performance. Where the fully cooperative approach often results in a system performance that is similar to the centralised solution, the computational complexity of the downstream cooperative algorithm is low as it only scales with the size of the subsystems.

### 2-4 Mixed integer optimisation

The combination of both discrete and continuous control inputs with a nonlinear prediction model yields an MINLP optimisation problem that has to be solved within every controller sampling interval. The class of MINLP problems is one of the most challenging types of optimisation problems to solve as the problems are generally classified as NP-hard [37].
means in practice that the search space, and therefore generally the computational time needed to solve the problem, increases exponentially with the size of the problem (i.e. increases with the number of control measures and the control horizon that is considered in the optimisation).

The remainder of this section is structured as follows. Firstly, different approaches and relaxations for solving the MINLP problems are outlined. With some of these approaches, the MINLP problems can be transformed into discrete optimisation problems, nonlinear non-convex continuous optimisation problems or combinations of those. Therefore, methods for solving these types of optimisation problems are described. Finally, a conclusion is made on the approach that will be used for solving the MINLP problems in this work.

2-4-1 Approaches

Evaluating potential solutions

The most obvious approach for solving an MINLP problem is to evaluate all potential solutions. This can be done by evaluating a nonlinear non-convex continuous optimisation problem for every feasible VSL combination to find matching RM rates. The matching VSLs and RM rates that yield the lowest objective function value are the optimal solutions to the problem. However, this approach is generally impractical, as the number of potential VSL combinations that have to be evaluated grows exponentially with the size of the problem.

A bi-level optimisation [10] approach can be taken to reduce the number of potential solutions, where a higher level determines for which high potential VSLs input sequences the continuous optimisation has to be evaluated. However, this approach can potentially result in the loss of global optimality and the computational load is high.

Heuristic search algorithms

Some heuristic search algorithms can be used to find good (i.e. near-optimal) solutions to MINLP problems at a reasonable computational cost. However, they cannot guarantee a feasible nor globally optimal solution [48].

Genetic Algorithm (GA) [48] is a heuristic search algorithm that is known to effectively deal with MINLP problems [23]. GAs are based on the principles of natural genetics and natural selection. They use basic elements of natural genetics in the search procedure, like reproduction, crossover and mutation [48]. Other search algorithms that can deal with MINLP problems are simulated annealing and artificial neural networks [48].

Relaxation on control inputs

The optimisation problem can be simplified by assuming that the VSLs are continuous decision variables. By doing so, a nonlinear non-convex continuous optimisation problem has to be solved, instead of an MINLP problem. Since the VSLs are only allowed to take values from a prescribed discrete set, some approaches round the VSLs to acceptable values [4]. However, making this relaxation entails in many cases a large loss in performance [23].
Relaxation on prediction model

Propositions have been made to rewrite CTM and METANET as MLD models [1, 36]. The nonlinearities of the original models are relaxed by adding equalities, inequalities and auxiliary variables to the models [3]. The resulting optimisation problem is a Mixed Integer Linear Programming (MILP) problem, for which efficient solving techniques exist [12]. However, the relaxation results in a performance loss [36] because the prediction model is less accurate.

Alternating optimisation

Alternating optimisation is a relaxation method that has shown great results for freeway traffic control [23]. It decomposes the MINLP problem in two smaller optimisation problems: a discrete optimisation problem for finding VSLs and a non-convex nonlinear continuous optimisation problem for finding RM rates. The method is an iterative optimisation scheme where the discrete optimisation is solved by using RM rates that are found in the previous continuous optimisation as constant variables, and subsequently, the continuous optimisation is solved by using the VSLs that are found in the previous iteration as constant variables. The time-shifted input sequence of the previous controller sample is used as a starting point for the iteration in [23].

2-4-2 Discrete optimisation

A problem is classified as a discrete optimisation problem if all decision variables are integers or discrete variables. To map all possible input combinations, a search tree can be generated [48]. By evaluating the objective function for every tree branch, the global optimum to the problem can be found [23]. However, the size of the tree grows exponentially with the number of decision variables. Therefore, finding the global optimum to a discrete optimisation problem by evaluating all feasible inputs is classified as NP-hard. Subsequently, for freeway traffic control, this method is only applicable for a small number of VSLs and small horizons. For larger problems, smart search strategies exist that only evaluate specific inputs for the problem [18]. By doing so, these algorithms reduce the computational complexity but often lose global optimality. Heuristic search algorithms, which are described in more detail in Section 2-4-1, can also solve discrete optimisation problems [48]. However, these cannot prove (nor often provide) a globally optimal solution.

2-4-3 Nonlinear non-convex continuous optimisation

Finding continuous inputs in combination with a nonlinear non-convex prediction model yields a nonlinear non-convex optimisation problem. There are no optimisation methods that can guarantee a globally optimal solution for such a problem in finite time [48]. However, Sequential Quadratic Programming (SQP) [48] is known to be an effective optimisation method for solving nonlinear non-convex continuous optimisation problems. Although the method cannot guarantee a global optimum, adequately combining SQP with a multi-start approach can significantly increase the possibility of attaining a globally optimal solution.
With a multi-start approach, multiple optimisation problems are evaluated with different initial points. For a detailed description of SQP and other gradient-based methods that are capable of solving the problem, the reader is referred to [48].

Some heuristic search algorithms can also deal with nonlinear non-convex continuous optimisation problems, which are discussed in Section 2-4-1.

2-4-4 Conclusions

An alternating optimisation scheme will be used to relax the MINLP optimisation problems because it has shown great results when VSLs and RM installations are used as control measures. SQP will be used for solving the continuous optimisation problems of the alternating optimisation schemes because it is likely that global optima are attained by properly combining SQP with a multi-start approach. For the discrete part of the alternating optimisation scheme, two methods will be used. For the discrete optimisation problems of the distributed and decentralised algorithms, all feasible solutions of the solution tree will be evaluated. The method guarantees global optima to the sub-problems and is implementable, since their solution space is relatively small. A GA will be used for the discrete optimisations of the centralised algorithm, as the solution space is much larger.

2-5 Conclusions

In this chapter, the relevant literature on the application of MPC to large freeway networks has been discussed. In the first section, different freeway traffic models have been outlined. METANET will be used as the prediction model, as it has proven a good trade-off between computational complexity and model accuracy. In the second section, the different components of the objective functions have been outlined. The objective functions in this work will include three terms: the TTS for the reduction of traffic congestion, a term that penalises large and frequent fluctuations in metering rates and a soft constraint term for the queue lengths at the on-ramps. In the third section, distributed architectures have been compared to the more conventional decentralised and centralised schemes. A fully cooperative and downstream cooperative architecture will be used for the distributed algorithms, since they offer a good trade-off between computational complexity and system performance. In the last section, approaches for solving the MINLP optimisation problems have been outlined. An alternating optimisation scheme will be used to solve the MINLP problems, as it has shown great results when VSLs and RM installations are used as control measures. SQP in combination with a multi-start approach will be used to solve the continuous optimisation problems. The discrete optimisation problems of the distributed and decentralised algorithms will be solved by evaluating all feasible solutions in the search tree, whilst a GA will be used to solve the discrete optimisation problems of the centralised algorithm.
Chapter 3

Distributed MPC for Large Freeway Networks using Alternating Optimisation

This chapter proposes two novel MPC algorithms that use a distributed control architecture and an alternating optimisation scheme for coordinated control of RM installations and VSLs on large freeway networks. The algorithms are formalised using the notation and modelling concepts of METANET. If the reader is not familiar with those concepts, it is advisable to first read Appendix B. The chapter is structured as follows. Section 3-1 describes the control problem that the algorithms have to solve. Then, Section 3-2 describes and formalises both algorithms. Finally, Section 3-3 concludes the chapter.

3-1 Freeway network

An arbitrary long freeway network is considered that is modelled by \( N_{\text{all}} \) segments \( i \in I_{\text{all}} \). All segments \( i \in I_r \) contain an on-ramp and all segments \( i \in I_{\text{off}} \) contain an off-ramp. All on-ramps have an RM installation that can control the number of vehicles that enter the network. Furthermore, the segments \( i \in I_{\text{VSL}} \) contain a matrix sign that can display speed limits with values that are in the discrete set \( \text{VSL}_{\text{set}} \). The network is subject to a mainstream demand \( d_0 \) and on-ramp demands \( d_i, \forall i \in I_r \).

Moreover, the following assumptions are made for the control of the network:

1. The measurements of the states \( \rho_i(k), v_i(k), \forall i \in I_{\text{all}} \) and \( w_i(k), \forall i \in I_r \) are available at every controller sampling time \( k_c = \frac{k}{N_r} \).

2. The measurements of the demands are available at every controller sampling time.

3. The agents that control the network receive all necessary measurements of the states and demands at every controller sampling time without delay.
4. All agents can communicate their decision variables to all the other agents in the network without delay.

The following section proposes two MPC algorithms for coordinated control of the RM installations and VSLs in this network to minimise the traffic congestion.

### 3-2 Control algorithms

#### 3-2-1 Overview

The previous section described an arbitrary long freeway network with an arbitrary number of VSLs and RM installations that can be controlled. This section proposes two MPC algorithms, FC-A-MPC and DC-A-MPC, for coordinated control of those control measures to minimise the traffic congestion in the network. Both algorithms use a distributed control architecture and an alternating optimisation scheme to reduce the computational complexity of the control problem. Schematics of both algorithms are depicted in Figure 3-1 and 3-2. The freeway is partitioned in subsystems, each controlled by a local agent. The agents use the alternating optimisation scheme to decompose the MINLP problems that are associated with determining discrete VSLs and continuous RM rates into continuous optimisation problems and discrete optimisation problems. Subsequently, the agents iterate between solving the discrete and continuous optimisation problems, while they communicate intermediate solutions to other agents. The algorithms differ in communication and optimisation protocols. The remainder of this section describes the details of both algorithms. Firstly, the distributed architectures of both algorithms are described. Subsequently, operational constraints on the VSLs to reduce the computational complexity of the problem and improve driver safety and comfort are discussed. Then, the alternating optimisation problems are formulated for both algorithms. Finally, the communication and optimisation protocols of both algorithms are formalised.

![Figure 3-1: An illustration of the communication and optimisation protocols of FC-A-MPC. All agents cooperative with each other to coordinate the control measures of the network.](image-url)
3-2-2 Distributed architecture

Both proposed algorithms use a distributed architecture to control the freeway network. Hence, the freeway network is partitioned into $N_{\text{sub}}$ subsystems, where each subsystem contains an arbitrary number of VSLs and RM installations. Subsequently, $I_{\text{all}}$ is partitioned for every subsystem $s$ in $I_s$, such that all segments in $I_s$ are part of subsystem $s$. Similarly, $I_r$, $I_{\text{VSL}}$, $I_{\text{off}}$ are partitioned in $I_{s\,r}$, $I_{s\,\text{VSL}}$, $I_{s\,\text{off}}$ for every subsystem $s$. Associated with the subsystems are $N_{\text{sub}}$ MPC agents, that determine local control inputs for their respective subsystem.

**Remark.** To improve readability, a superscript is used for the notation of a partitioned subset. This should not be confused with an exponential value.

**FC-A-MPC**

With FC-A-MPC, all the agents cooperate to coordinate the control measures of the network [25]. Hence, all agents optimise one global objective function, describing the performance of the comprehensive network. However, the agents only optimise this objective function for their own respective control inputs, while they consider the RM rates and VSLs of the other agents as constant variables in the optimisation. A parallel and iterative scheme is implemented to coordinate the control inputs of the network: once all agents have solved their optimisation problem for local control inputs, they communicate the solutions to the other agents and, subsequently, repeat solving the optimisation problem with the updated solutions of the other agents. This process is repeated for $n_{\text{dist}}$ iterations. Since all agents optimise one global objective function for local control inputs, no coupling between objective functions is necessary.

The global objective function at time $k_c$ for agent $s$ is derived from the sub-objectives described in Section 2-2:
\[ J^s(k_c) = \sum_{k=Mk_c}^{M(k_c+Np)} \left[ T_c \left( \sum_{i \in I_{all}} \rho_i(k)L_i \lambda + \sum_{i \in I_s} w_i(k) \right) + \right. \\
\left. \zeta_w \left( \sum_{i \in I_s} \left( \max(w_i(k) - w_{\text{max}}, 0) \right)^2 \right) \right] + \\
\zeta_r \sum_{j=1}^{Np-1} \left[ \left( \sum_{i \in I_{s+r}} \left( U_{s+r}^i(k_c + j) - U_{s+r}^i(k_c + j - 1) \right) \right)^2 \right], \tag{3-1} \]

where the first term minimises the congestion in the network, the second term imposes soft constraints of \( w_{\text{max}} \) on the queue lengths at the on-ramps and the last term prevents frequent and large fluctuations in the RM signals.

**DC-A-MPC**

As opposed to FC-A-MPC, with DC-A-MPC the agents only cooperate with their downstream neighbouring agent to coordinate the control measures of the network [19]. Hence, all agents optimise a local objective function that contains the states of their own subsystem and the states of the downstream subsystem. However, the agents only optimise this objective function for their own respective control inputs, while they consider the RM rates and VSLs of the downstream agent as constant variables in the optimisation. A parallel and iterative scheme is implemented to coordinate the control inputs of the network: once all agents have solved their optimisation problem for local control inputs, they communicate the solutions to their respective upstream neighbour and, subsequently, repeat solving the optimisation problem with the updated solutions of their downstream neighbour. This process is repeated for \( n_{\text{dist}} \) iterations. As opposed to FC-A-MPC, coupling between the local objective functions is necessary with DC-A-MPC. However, the coupling is straightforward due to the spatial nature of freeway networks, as the dynamics of the different subsystems are described by the same equations. Hence, the flow and the average speed of the last segment of the upstream subsystem and the density of the first segment of the downstream subsystem are used to describe the boundary conditions of a subsystem, instead of the boundary conditions described in Appendix B.

The local objective function at time \( k_c \) for agent \( s \) is derived from the sub-objectives described in Section 2-2:

\[ J^s(k_c) = \sum_{k=Mk_c}^{M(k_c+Np)} \left[ T_c \left( \sum_{i \in \{I_s, I_{s+1}\}} \rho_i(k)L_i \lambda + \sum_{i \in \{I_{s+r}, I_{s+r+1}\}} w_i(k) \right) + \\
\zeta_w \left( \sum_{i \in \{I_{s+r}, I_{s+r+1}\}} \left( \max(w_i(k) - w_{\text{max}}, 0) \right)^2 \right) \right] + \\
\zeta_r \sum_{j=1}^{Np-1} \left[ \left( \sum_{i \in I_{s+r}} \left( U_{s+r}^i(k_c + j) - U_{s+r}^i(k_c + j - 1) \right) \right)^2 \right], \tag{3-2} \]

where the first term minimises the congestion in the considered part of the network, the second term imposes soft constraints of \( w_{\text{max}} \) on the queue lengths at the on-ramps in the
considered part of the network and the last term prevents frequent and large fluctuations in the RM signals.

3-2-3 Operational constraints

The fluctuations in metering rates are penalised by a term in the objective functions (in Equation (3-1) and (3-2)). Similarly, it is necessary to avoid fluctuations in the VSLs to improve driver safety and comfort [4, 38]. This is done with hard constraints because the VSLs are discrete decision variables. Hence, the hard constraints reduce the size of the solution space of the VSLs and, therefore, reduce the complexity of the optimisation problems. Two types of constraints on the VSLs are considered in this work. The first constraint allows the VSLs to maximally change $\eta_t$ per controller sample [4, 38]:

$$\left| U_{VSL}^i(k_c) - U_{VSL}^i(k_c + 1) \right| \leq \eta_t, \quad \forall i \in I_{VSL}.$$  \hspace{1cm} (3-3)

The second constraint allows the VSLs that are on two consecutive freeway segments to maximally differ $\eta_d$ from each other [4, 38]:

$$\left| U_{VSL}^i(k_c) - U_{VSL}^{i+1}(k_c) \right| \leq \eta_d, \quad \forall i : \{i, i + 1\} \subseteq I_{VSL}.$$  \hspace{1cm} (3-4)

3-2-4 Alternating optimisation

The agents use an alternating optimisation scheme [23] to solve their optimisation problems. With the alternating optimisation scheme, the MINLP optimisation problem is decomposed into one nonlinear non-convex continuous optimisation problem to find RM rates, and one discrete optimisation problem to find VSLs. Hence, the comprehensive solution space is reduced into one discrete solution space for the VSLs, and one continuous solution space for the RM rates. In the continuous optimisation problems, the VSLs are considered to be constant variables, and vice versa, in the discrete optimisation problems, the RM rates are considered to be constant variables. The optimisation problems are solved one after another for $n_{alt}$ iterations. For every new iteration, the RM rates that are found in the previous continuous optimisation problem are used as updated constant variables in the discrete optimisation. Likewise, the VSLs that are found in the previous discrete optimisation problem are used as updated constant variables in the continuous optimisation. The continuous optimisation problems and the discrete optimisation problems for both FC-A-MPC and DC-A-MPC are formalised in the remainder of this section.

**FC-A-MPC**

Every agent $s$ optimises the continuous part $\bar{J}_{con}^s$ of the global objective function to find the optimal metering rates $\bar{U}_r^s$ in its subsystem over the prediction horizon $N_p$. Hence, at time $k_c$, $\bar{U}_r^s(k_c)$ contains the optimal metering rates for every RM installation in subsystem $s$ for every sample in the horizon $\{k_c, k_c + 1, \ldots, k_c + N_p - 1\}$. The terms $\bar{U}_r^s$ and $\bar{U}_{VSL}^s$ contain all intermediate solutions that are considered constant in the optimisation problem. Hence, the RM rates of the other agents are included in $\bar{U}_r^s$ and are constant variables in the
Similarly, every agent and $\zeta$ space of the RM rates, the continuous optimisation problem for agent $s$ is updated with the optimal control inputs that have been found by the other agents.

The continuous optimisation problem for agent $s$ is formulated as:

$$
\min_{\dot{U}_{s}^{\text{con}}(k_c)} J_{\text{con}}^{s}(k_c)
$$

subject to:

$$
x(k_c + l + 1) = f_M((x(k_c + l), \hat{U}_{s}^{l}(k_c + l), U_{s}^{c}(k_c + l), \hat{U}_{\text{VSL}}^{c}(k_c + l), d(k_c + l)),
$$

$$
x(k_c) = x_k,
$$

$$
d(k_c + l) = d_k,
$$

$$
U_{s}^{l}(k_c + l) \in U_{s}, \forall i \in I_{s},
$$

$$
U_{s}^{c}(k_c + j) = U_{s}^{c}(k_c + N_c - 1), \forall i \in I_{s},
$$

for $j \in \{N_u, \ldots, N_p - 1\}$,

for $l \in \{0, 1, \ldots, N_p - 1\},
$$

where $\dot{J}_{\text{con}}^{s}(k_c)$ is the continuous part of the global objective function defined in Equation (3-1):

$$
\dot{J}_{\text{con}}^{s}(k_c) = \sum_{k=0}^{M(k_c + N_p)} \left[ T_{c} \left( \sum_{i \in I_{\text{all}}} \rho_{i}(k)L_{i}\lambda + \sum_{i \in I_{s}} w_{i}(k) \right) + \zeta_{w} \left( \sum_{i \in I_{s}} \left( \max(w_{i}(k) - w_{\text{max}}, 0) \right)^{2} \right) \right]^{2} + \zeta_{r} \sum_{j=1}^{N_{p}} \left[ \left( \sum_{i \in I_{s}} \left( U_{s}^{c}(k_c + j) - U_{s}^{c}(k_c + j - 1) \right)^{2} \right) \right],
$$

and the future states $x$ of the comprehensive network are predicted by the system dynamics $f_M$ of METANET, $x_k$ are the measurements of the states at time $k_c$, $d$ are the demands of the comprehensive network, $d_k$ are the measurements of the demands at time $k_c$, $U_{s}$ is the input space of the RM rates, $N_u$ is the control horizon that simplifies the optimisation problem, and $\zeta_{w}$ and $\zeta_{r}$ are weighting terms.

Similarly, every agent $s$ optimises the discrete part $\dot{J}_{\text{dis}}^{s}$ of the global objective function to find the optimal speed limits $\hat{U}_{\text{VSL}}^{s}$ in its subsystem over prediction horizon $N_p$. Hence, at time $k_c$, $\hat{U}_{\text{VSL}}^{s}(k_c)$ contains all the optimal VSLs of subsystem $s$ for every sample in the horizon $\{k_c, k_c + 1, \ldots, k_c + N_p - 1\}$. The speed limits of the other agents are included in $\hat{U}_{\text{VSL}}^{s}$ and are constant variables in the optimisation. All metering rates $U_{s}^{c}$, including both the metering rates of agent $s$ and all the other agents, are constant variables in the optimisation. After every discrete optimisation, $\hat{U}_{\text{VSL}}^{s}$ is updated with optimal speed limits $\hat{U}_{\text{VSL}}^{s}$. After every
distributed iteration, $\hat{U}_s^r$ and $\hat{U}_{VSL}^s$ are updated with the optimal control inputs that have been found by the other agents.

The discrete optimisation problem for agent $s$ is formulated as:

$$\min_{\bar{U}_{VSL}^s(k_c)} \bar{J}_{\text{dis}}^s(k_c)$$

subject to:

$$x(k_c + l + 1) = f_M((x(k_c + l), \bar{U}_{VSL}^s(k_c + l), \hat{U}_s^r(k_c + l), \hat{U}_{VSL}^s(k_c + l), d(k_c + l)),$$

$$x(k_c) = x_k,$$

$$d(k_c + l) = d_k,$$

$$U_{VSL}^i(k_c + l) \in \mathcal{U}_{VSL}, \forall i \in I_s^v,$$

$$U_{VSL}^i(k_c + j) = U_{VSL}^i(k_c + N_v - 1), \forall i \in I_s^v,$$

for $j \in \{N_v, \ldots, N_p - 1\}$,

for $l \in \{0, 1, \ldots, N_p - 1\}$,

$$(3-9)$$

where $\bar{J}_{\text{dis}}^s(k_c)$ is the discrete part of the global objective function defined in Equation (3-1):

$$\bar{J}_{\text{dis}}^s(k_c) = \sum_{k=M(k_c+N_p)}^{M(k_c+k)} \left[ T_c \left( \sum_{i \in I_{\text{all}}} \rho_i(k) L_i \lambda + \sum_{i \in I_c} w_i(k) \right) + \zeta_w \left( \sum_{i \in I_c} \left( \max(w_i(k) - w_{\text{max}}, 0) \right)^2 \right) \right],$$

$$(3-10)$$

and $\mathcal{U}_{VSL}$ is the input space of the VSLs.

Remark. It has to be observed that $U_{VSL}^i(k_c)$ is the VSL on segment $i$ at time $k_c$, while $\bar{U}_{VSL}^s(k_c)$ is the set of optimal inputs at time $k_c$ for all the VSLs that are in subsystem $s$ over the horizon $N_p$. $U_{VSL}^i(k_c)$ and $\bar{U}_{VSL}^s(k_c)$ are defined similarly.

DC-A-MPC

The optimisation problems in the DC-A-MPC scheme are very similar to the optimisation problems in the FC-A-MPC scheme, containing a few subtle differences due to the difference in cooperativeness. For completeness, the optimisation problems are described and formulated here.

Every agent $s$ optimises the continuous part of the local objective function $J_{\text{con}}^s$ to find the optimal metering rates $\bar{U}_s^r$ in its subsystem over the prediction horizon $N_p$. Hence, at time $k_c$, $\bar{U}_s^r(k_c)$ contains the optimal metering rates for every RM installation in subsystem $s$ for every sample in the horizon $\{k_c, k_c + 1, \ldots, k_c + N_p - 1\}$. The terms $\bar{U}_s^r$ and $\hat{U}_{VSL}^s$ contain all intermediate solutions that are considered constant in the optimisation problem. Hence, the RM rates of agent $s + 1$ are included in $\bar{U}_s^r$ and are constant variables in the optimisation. The speed limits of agent $s$ and agent $s + 1$, $\hat{U}_{VSL}^s$, are constant variables in the optimisation.

Master of Science Thesis

Uglješa Todorović
After every continuous optimisation, $\hat{U}_i^s$ is updated with new optimal metering rates $\hat{U}_i^s$. After every distributed iteration, $\hat{U}^s$ and $\hat{U}_{\text{VSL}}^s$ are updated with the optimal control inputs that have been found by agent $s + 1$.

The continuous optimisation problem of agent $s$ is formulated as:

$$\min_{U_i^s(k_c)} J^s_{\text{con}}(k_c)$$

subject to:

$$x(k_c + l + 1) = f_M((x(k_c + l), \hat{U}_i^s(k_c + l), \hat{U}_{\text{VSL}}^s(k_c + l), d(k_c + l)), x(k_c) = x_k,$$

$$d(k_c + l) = d_k,$$

$$U_i^s(k_c + l) \in \mathcal{U}_i, \forall \in I^s_i,$$

$$U_i^s(k_c + j) = U_i^s(k_c + N_c - 1), \forall \in I^s_i,$$

for $j \in \{N_u, \ldots, N_p - 1\},$

for $l \in \{0, 1, \ldots, N_p - 1\},$

$$J^s_{\text{con}}(k_c)$$

where $J^s_{\text{con}}(k_c)$ is the continuous part of the local objective function defined in Equation (3-2):

$$J^s_{\text{con}}(k_c) = \sum_{k=M(k_c+N_p)}^{M(k_c+N_p)} \left[ T_c \left( \sum_{i \in \{ I_x, I_{x+1} \}} \rho_i(k) L_i \lambda + \sum_{i \in \{ I_r^s, I_{r+1}^s \}} w_i(k) \right) + \zeta_w \left( \sum_{i \in \{ I_r^s, I_{r+1}^s \}} \left( \max(w_i(k) - w_{\text{max}}, 0) \right)^2 \right) \right] +$$

$$\zeta_r \sum_{j=1}^{N_r-1} \left[ \left( \sum_{i \in I_r^s} (U_i^s(k_c + j) - U_i^s(k_c + j - 1)) \right)^2 \right],$$

the future states $x$ of subsystem $s$ and $s + 1$ are predicted by the system dynamics $f_M$ of METANET and $d$ contains the demands at all the on-ramps in subsystem $s$ and $s + 1$.

Similarly, every agent $s$ optimises the discrete part of the local objective function $J^s_{\text{dis}}$ to find the optimal speed limits $\hat{U}_{\text{VSL}}^s$ in its subsystem over prediction horizon $N_p$. Hence, at time $k_c$, $\hat{U}_{\text{VSL}}^s(k_c)$ contains all the optimal VSLs of subsystem $s$ for every sample in the horizon $\{k_c, k_c + 1, \ldots, k_c + N_p - 1\}$. The speed limits of the downstream agent are included in $\hat{U}_{\text{VSL}}^s$ and are constant variables in the optimisation. The metering rates of agent $s$ and agent $s + 1$, $\hat{U}_i^s$, are constant variables in the optimisation. After every discrete optimisation, $\hat{U}_{\text{VSL}}^s$ is updated with optimal speed limits rates $\hat{U}_{\text{VSL}}^s$. After every distributed iteration, $\hat{U}_i^s$ and $\hat{U}_{\text{VSL}}^s$ are updated with the optimal control inputs that have been found by agent $s + 1$.

The discrete optimisation problem of agent $s$ is formulated as:
\[
\min_{U_{\text{VSL}}(k_c)} J_{\text{dis}}^*(k_c)
\]

subject to:

\[
x(k + l + 1) = f_M((x(k_c + l), \bar{U}_{\text{VSL}}(k_c + l), \hat{U}_c(k_c + l), \dot{U}_{\text{VSL}}(k_c + l), d(k_c + l)),
\]

\[
x(k_c) = x_k,
\]

\[
d(k_c + l) = d_k,
\]

\[
U_{\text{VSL}}(k_c + l) \in \mathcal{U}_{\text{VSL}}, \forall i \in I_{\text{VSL}},
\]

\[
U_{\text{VSL}}^i(k_c + j) = U_{\text{VSL}}^i(k_c + N_c - 1), \forall i \in I_{\text{VSL}},
\]

for \( j \in \{N_u, \ldots, N_p - 1\} \),

for \( l \in \{0, 1, \ldots, N_p - 1\} \),

(3-15)

where \( J_{\text{dis}}^*(k_c) \) is the discrete part of the local objective function defined in Equation (3-2):

\[
J_{\text{dis}}^*(k_c) = \sum_{k = M k_c}^{M(k_c + N_p)} \left[ T_c \left( \sum_{i \in I_{\text{VSL}}^i} \rho_i(k) L_i \lambda + \sum_{i \in I_{\text{VSL}}^i} w_i(k) \right) + \right.
\]

\[
\zeta_w \left( \sum_{i \in I_{\text{VSL}}^i} \left( \max(w_i(k) - w_{\text{max}}, 0) \right)^2 \right). \quad (3-16)
\]

### 3-2-5 Initialisation and stopping criteria

As explained before, the agents consider the control inputs of other agents as constant variables in their optimisation problems. Once all agents have solved their optimisation problem, these constant variables are updated with the new optimal values. However, for the first distributed iteration, initial values for these constant variables are necessary to initialise this iterative optimisation scheme. For these initial values, the agents use the time-shifted VSLs of the previous controller sample [23].

The agents use two stopping criteria to terminate the optimisation scheme:

1. \( \text{CT} \geq t_{\text{term}} \): If the Computational Time (CT) of the agents becomes larger than \( t_{\text{term}} \), the optimisation is terminated. Hence, the agents solve the optimisation problems within the controller sampling intervals if \( t_{\text{term}} = T_c \).

2. \( n_{\text{dist}} \): The optimisation problems of the final distributed iteration \( n_{\text{dist}} \) are solved. To only use \( t_{\text{term}} \) as stopping criterion, \( n_{\text{dist}} \) is set to \( n_{\text{dist}} = \infty \).

After the optimisation scheme is terminated, the first time samples of the inputs that have been found in the distributed iteration that yield the best global objective function value are used as inputs for the freeway network.
3-2-6 Algorithm formulation

Both FC-A-MPC and DC-A-MPC use the same iterative scheme, described in Algorithm 1. Since the algorithms differ in cooperativeness, the variables in Algorithm 1 are defined differently for both algorithms. The communication and optimisation protocols of both algorithms are described in the remainder of this section.

**FC-A-MPC**

The constant variables \( \hat{U}_s^r \) and \( \hat{U}_s^{VSL} \) of agent \( s \) contain the intermediate inputs of agent \( s \) (local part) and the intermediate inputs of all the other agents of the network (nonlocal part). To initialise the algorithm, the control inputs of the previous controller sample, \( U_{VSL}^{prev} \) and \( U_{r}^{prev} \), are used as constant variables. Every agent solves the continuous optimisation problem in Equation (3-5) and the discrete optimisation problem in Equation (3-8) \( n_{alt} \) times for every distributed iteration. The local part of \( \hat{U}_s^r \) is updated with \( \bar{U}_s^r \) every time agent \( s \) solves the continuous optimisation problem. Similarly, the local part of \( \hat{U}_s^{VSL} \) is updated with \( \bar{U}_s^{VSL} \) every time agent \( s \) solves the discrete optimisation problem. After \( n_{alt} \) iterations, the agents communicate their optimal inputs \( \bar{U}_s^r \) and \( \bar{U}_s^{VSL} \) to the other agents and receive the intermediate optimal inputs \( U_{r}^{other} \) and \( U_{VSL}^{other} \) of the other agents in the network. Subsequently, all agents update the nonlocal parts of \( \hat{U}_s^r \) and \( \hat{U}_s^{VSL} \) with \( \bar{U}_r^{other} \) and \( \bar{U}_{VSL}^{other} \), and resolve to the next distributed iteration. The algorithm is terminated after \( n_{dist} \) distributed iterations or when the CT becomes larger than \( t_{term} \). After every distributed iteration, the local inputs that have been found are stored. Finally, the global objective function is evaluated for the inputs that have been found in every distributed iteration. The first time samples of the inputs that yield the best global objective function value are sent to the network as control inputs.

**DC-A-MPC**

The constant variables \( \hat{U}_r^s \) and \( \hat{U}_r^{VSL} \) of agent \( s \) contain the intermediate inputs of agent \( s \) (local part) and the intermediate inputs of the downstream agent \( s + 1 \) (nonlocal part). To initialise the algorithm, the control inputs of the previous controller sample, \( U_{VSL}^{prev} \) and \( U_{r}^{prev} \) are used as constant variables. Every agent solves the continuous optimisation problem in Equation (3-11) and the discrete optimisation problem in Equation (3-14) \( n_{alt} \) times for every distributed iteration. The local part of \( \hat{U}_r^s \) is updated with \( \bar{U}_r^s \) every time agent \( s \) solves continuous optimisation problem. Similarly, the local part of \( \hat{U}_r^{VSL} \) is updated with \( \bar{U}_r^{VSL} \) every time agent \( s \) solves the discrete optimisation problem. After \( n_{alt} \) iterations, the agents communicate their optimal inputs \( \bar{U}_r^s \) and \( \bar{U}_r^{VSL} \) to their upstream neighbouring agent and receive the optimal inputs \( U_{r}^{other} \) and \( U_{VSL}^{other} \) of their downstream neighbouring agent. Subsequently, all agent update the nonlocal parts of \( \hat{U}_r \) and \( \hat{U}_{VSL} \) with \( U_{r}^{other} \) and \( U_{VSL}^{other} \), and resolve to the next distributed iteration. The algorithm is terminated after \( n_{dist} \) distributed iterations or when the CT becomes larger than \( t_{term} \). After every distributed iteration, the local inputs that have been found are stored. Finally, the global objective function is evaluated for the inputs that have been found in every distributed iteration. The first time samples of the inputs that yield the best global objective function value are sent to the network as control inputs.
Algorithm 1: Top level communication and optimisation protocol of FC-A-MPC and DC-A-MPC for agent $s$.

**Input:** $U_{\text{prev}}^\text{VSL}, U_{\text{prev}}^r, t_{\text{term}}, n_{\text{dist}}, n_{\text{alt}}$

**Output:** $\bar{U}_s^r, \bar{U}_s^\text{VSL}$

Start timer

Set initial values

$\hat{U}_s^\text{VSL} := U_{\text{prev}}^\text{VSL}$

$\hat{U}_s^r := U_{\text{prev}}^r$

Start iterative scheme

**while** $t_{\text{timer}} \leq t_{\text{term}}$ **do**

**for** $\text{dist} = 1$: $n_{\text{dist}}$ **do**

**for** $\text{alt} = 1$: $n_{\text{alt}}$ **do**

Solve continuous optimisation for $\hat{U}_s^r$

Update local part of $\hat{U}_s^r$ with $\bar{U}_s^r$

Solve discrete optimisation for $\hat{U}_s^\text{VSL}$

Update local part of $\hat{U}_s^\text{VSL}$ with $\bar{U}_s^\text{VSL}$

end

end

Store $\bar{U}_s^r$ and $\bar{U}_s^\text{VSL}$

Communicate $\bar{U}_s^r$ and $\bar{U}_s^\text{VSL}$ to other agents

Receive optimal inputs $\bar{U}_{\text{other}}^r$ and $\bar{U}_{\text{other}}^\text{VSL}$ of other agents

Update nonlocal part of $\hat{U}_s^r$ with $\bar{U}_{\text{other}}^r$

Update nonlocal part of $\hat{U}_s^\text{VSL}$ with $\bar{U}_{\text{other}}^\text{VSL}$

end

end

3-3 Conclusions

Two MPC algorithms, FC-A-MPC and DC-A-MPC, have been proposed for coordinated control of continuous RM rates and discrete VSLs for large freeway networks. Both algorithms use a distributed architecture and an alternating optimisation scheme to reduce the computational complexity of the problem. In the first section, the control problem that the algorithms have to solve has been formalised. In the second section, the different aspects of both algorithms have been discussed. Firstly, the distributed architectures of both algorithms have been outlined. The agents controlled by FC-A-MPC are fully cooperative and, hence, optimise a global objective function. The agents controlled by DC-A-MPC are downstream cooperative and, hence, optimise a local objective function. Secondly, operational constraints have been imposed on the input space of the VSLs to reduce the computational complexity of the optimisation problems and improve driver safety and comfort. Subsequently, the alternating optimisation scheme has been outlined and the optimisation problems that the agents have to solve with both algorithms have been formulated. Then, stopping criteria and initialisation of both algorithms have been discussed. Finally, the communication and optimisation protocols of both algorithms have been discussed and formalised.
Chapter 4

Case Study and Comparison

The algorithms that are proposed in Chapter 3 are evaluated in this chapter. This is done by comparing their performance to more conventional MPC algorithms in a case study. The chapter is structured as follows. Section 4-1 discusses the details of the case study, including the layout of the freeway network that is used, the performance criteria and the no-control network response. Section 4-2 discusses the design choices that are made on the control parameters to fit the proposed algorithms to the case study. Then, the case study is described in two sections. Section 4-3 describes the first part of the case study, where the effects on the system performance by relaxing the VSLs to be continuous decision variables instead of discrete decision variables are investigated. Section 4-4 describes the second part of the case study, where the proposed algorithms are compared to the more conventional centralised and decentralised algorithms. Lastly, the chapter is concluded in Section 4-5.

4-1 Case study details

This section discusses the details of the case study. Firstly, the layout of the freeway network is described and details are given on the traffic circumstances in the simulations. Then, the performance criteria that are used for evaluating the control algorithms are discussed. Subsequently, specifications of the computer on which the simulations are conducted are given. Lastly, the no-control network response is investigated to serve as a benchmark.

4-1-1 Freeway network and simulation

The freeway network from [24] is used to evaluate the performance of the proposed algorithms, but is slightly modified to include RM installations and is depicted in Figure 4-1. METANET is used to simulate this network, whose system dynamics are described in Appendix B.

The network has a length of 30 km, is partitioned into $N_{all} = 24$ segments in set $I_{all}$, contains six VSLs on segments $I_{VSL} = \{2,3,9,10,16,17\}$, contains three RM installations at the on-ramps on segments $I_r = \{7,14,21\}$ and contains three off-ramps at segments $I_{off} = \{5,12,19\}$. The VSLs are allowed to take values from the discrete set $VSL_{net} = \{40,60,80,100\}$ km/h.
The control inputs are enumerated in the downstream direction to improve readability as illustrated in Figure 4-1.

**Table 4-1:** The splitting fractions of the vehicular flow at the three off-ramps. The fractions remain constant during the simulation.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_5$</td>
<td>0.21</td>
</tr>
<tr>
<td>$\beta_{12}$</td>
<td>0.26</td>
</tr>
<tr>
<td>$\beta_{19}$</td>
<td>0.02</td>
</tr>
</tbody>
</table>

The system parameters of the network are chosen identically to [24] and are summarised in Table 4-2. Hence, the only model parameter that differs per segment is the segment length. The segments that contain VSLs have a larger length, so that the speed limits have more influence on the overall traffic state of the network. A hypothetical traffic situation is simulated for 2.5 hours in the case study, which corresponds to $N_{\text{sim}} = 900$ samples with model sampling time $T_m = 10$ s. The demand profile and splitting fractions are chosen similarly to [24], but are modified so that larger traffic jams occur when no control is applied. The demand profile and splitting fractions are shown in Figure 4-2 and Table 4-1, respectively. The no-control system response is described in Section 4-1-4. To avoid spillback to hypothetical urban roads, soft constraints are imposed on the queue lengths on all three on-ramps of $w_{\text{max}} = 100$ veh.

**Table 4-2:** The parameter values of METANET describing the system dynamics of the freeway network that is used in the case study. The values are chosen identically to [24].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>1.867</td>
<td>$L_i {\forall i</td>
<td>1 \leq i \leq N_{\text{all}} \land i \notin I_{\text{VSL}}}$</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.1</td>
<td>$L_i {\forall i</td>
<td>i \in I_{\text{VSL}}}$</td>
</tr>
<tr>
<td>$N_{\text{all}}$</td>
<td>24</td>
<td>$w_{\text{max}}$</td>
<td>100 [veh]</td>
</tr>
<tr>
<td>$N_{\text{sim}}$</td>
<td>900</td>
<td>$C$</td>
<td>2000 [veh/h/lane]</td>
</tr>
<tr>
<td>$\delta$</td>
<td>0.0122</td>
<td>$\rho_{\text{crit}}$</td>
<td>33.5 [veh/km/lane]</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>3</td>
<td>$\rho_{\text{max}}$</td>
<td>180 [veh/km/lane]</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>40</td>
<td>$\mu_L$</td>
<td>20 [km²/h]</td>
</tr>
<tr>
<td>$\tau$</td>
<td>18 [s]</td>
<td>$\mu_H$</td>
<td>80 [km²/h]</td>
</tr>
<tr>
<td>$T_m$</td>
<td>10 [s]</td>
<td>$v_{\text{free}}$</td>
<td>102 [km/h]</td>
</tr>
</tbody>
</table>
4-1 Case study details

Figure 4-2: The demand at the mainstream origin and the on-ramps during the simulation.

4-1-2 Performance criteria

The goal of this thesis is to develop MPC algorithms that offer a good trade-off between system performance and computational complexity. Hence, two performance criteria are used to evaluate the control algorithms.

System performance

The TTS of the comprehensive freeway network is used to quantify the system performance of the algorithms:

\[
TTS = \sum_{k=1}^{N_{sim}} \left( T_m \left( \sum_{i \in I_{all}} \rho_i(k)L_i\lambda + \sum_{i \in I_r} w_i(k) \right) \right),
\]  \hspace{1cm} (4-1)

For convenience, this performance index is also expressed as a reduction \(TTS_{\text{red}}\) relative to the no-control case:

\[
TTS_{\text{red}} = \frac{TTS_{\text{nc}} - TTS}{TTS_{\text{nc}}} \cdot 100\%,
\]  \hspace{1cm} (4-2)

where \(TTS_{\text{nc}}\) is the TTS of the no-control case.

Computational complexity

To quantify the computational complexity of the algorithms, the Computational Time (CT) that is needed to determine the control inputs for every controller sample is investigated. For the CT of distributed algorithms, a summation is made of the computational times for the number of distributed iterations. The largest CT of all controller samples is denoted as \(CT_{\text{max}}\). Hence, a control algorithm is implementable in real-time if \(CT_{\text{max}}\) is smaller than the controller sampling time \(T_c\).
4-1-3 Computer specifications

All the simulations in this work are conducted on an HP ZBook Studio G4, containing an Intel Core i7 processor and 8GB of RAM. The simulations are evaluated in MATLAB R2018b, and all optimisation problems are solved with the Optimization Toolbox of MATLAB.

4-1-4 No-control system response

In an uncontrolled traffic setting, all ramps are open and all VSLs are equal to the maximum speed limits. Hence, the no-control VSLs for the network are:

\[ U_{VSL}^i(k) = 100 \text{ km/h} \quad \forall i \in I_{VSL}, \forall k, \quad (4-3) \]

and the no-control RM rates for the network are:

\[ U_r^i(k) = 1 \quad \forall i \in I_r, \forall k. \quad (4-4) \]

The consequence of not controlling the high traffic demand is a major traffic congestion in the network. Two large traffic jams occur that spread out over a large part of the freeway network. The traffic jams are illustrated in Figures 4-3 and 4-4. In the jams, the traffic density is high and the average speed low. During the traffic jams, not all vehicles can freely enter the network from on-ramps. Hence, queues originate at all three on-ramps, illustrated in Figure 4-5. The traffic congestion results in a no-control system performance of \( \text{TTS}_{nc} = 5986 \text{ veh-h} \).

![Figure 4-3: A heat map of the densities of all the segments of the network when no control is applied.](image-url)
This section discusses the design choices that are made on the control parameters to fit the proposed MPC algorithms to the case study. Firstly, the partitioning of the freeway network into subsystems is described. Then, the choices on the sampling time and MPC horizons are discussed. Subsequently, choices on the weights in the objective functions and constraints are discussed. Lastly, the optimal number of distributed and alternating iterations are determined.

### 4-2-1 System partitioning

The comprehensive freeway network is partitioned into three subsystems for the decentralised and distributed approaches. The partitioned network is illustrated in Figure 4-6 and the subsets of the subsystems are summarised Table 4-3. The network is partitioned in such a way that each subsystem contains one RM installation and two VSLs. Previous works have shown that the alternating optimisation problem can be solved accurately with this combination of actuators [23]. An alternative approach is to partition the system in such a
way, that every agent either contains VSLs or RM installations. Subsequently, the alternating optimisation scheme can be avoided.

![Diagram of freeway network partitioned into subsystems.](image)

**Figure 4-6:** An illustration of the freeway network partitioned into subsystems.

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Set name</th>
<th>Segments in set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I₁</td>
<td>{1, 2, 3, 4, 5, 6, 7}</td>
</tr>
<tr>
<td>2</td>
<td>I₂</td>
<td>{8, 9, 10, 11, 12, 13, 14}</td>
</tr>
<tr>
<td>3</td>
<td>I₃</td>
<td>{15, 16, 17, 18, 19, 20, 21, 22, 23, 24}</td>
</tr>
</tbody>
</table>

**Table 4-3:** A summary of the freeway segments per subsystem.

### 4-2-2 Sampling time and horizons

In the MPC framework, control inputs are calculated and updated every controller sampling time $T_c$. These inputs are determined for all samples in the prediction horizon $N_p$ by taking into account the network response of $N_p T_c$ time in the future. Hence, a trade-off has to be made between $N_p$ and $T_c$, such that $N_p T_c$ is large enough to capture sufficient system dynamics. Increasing $T_c$ allows the agents more time to solve their optimisation problems, but can result in a worse system performance as the control inputs are updated less frequently. The number of control inputs that have to be found scales linearly with $N_p$. Hence, increasing $N_p$ results in more difficult optimisation problems. To simplify the optimisation problems, the number of decision variables can be reduced by decreasing the control horizon $N_u$.

The majority of MPC approaches for freeway traffic control use a $T_c$ of at most a few minutes [4, 19, 24, 25]. All MPC algorithms in this case study use a sampling time of $T_c = 120$ s, as it allows a good trade-off between sufficient time for solving the optimisation problems and frequent enough control input updates. To match this sampling time, all algorithms use a prediction horizon of $N_p = 10$, so that the network response of 1200 s in the future is taken into account during the optimisation. Hence, at free-flow speed, the algorithms take into account the influences of vehicles over a spatial distance of 34 km, which is slightly larger than the network size. A control horizon of $N_u = 3$ is used to simplify the optimisation problems. The sampling time and MPC horizons are summarised in Table 4-4.
Table 4-4: The controller sampling time $T_c$, prediction horizon $N_p$ and control horizon $N_u$ that are used with all MPC algorithms in this case study.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_c$</td>
<td>120 [s]</td>
</tr>
<tr>
<td>$N_p$</td>
<td>10</td>
</tr>
<tr>
<td>$N_u$</td>
<td>3</td>
</tr>
</tbody>
</table>

4-2-3 Weights of objective functions and constraints

The objective functions (defined in Equation (3-1) and (3-2)) contain two weighting terms: a term $\zeta_r$ that weights the influence of the fluctuation in RM rates and a term $\zeta_w$ that weights the influence of the soft constraints relative to the TTS. Generally, the signal fluctuation term is used to avoid noisy RM rates. However, it is chosen to neglect this term to avoid a tedious tuning process and to make the comparison of the different MPC algorithms more straightforward. The RM rates in this work are not extremely noisy with $\zeta_r = 0$, because the controllers have a relatively large $T_c$. The soft constraint weights are chosen $\zeta_w = 10$, as this results in a good trade-off between queue length behaviour and system performance. Consequently, the algorithms do not violate the soft constraints by more than 10 % with this weight. A 10 % violation in queue constraints is considered acceptable in this work, since a hypothetical network is used without details on the urban roads surrounding the network. If less violation is desirable, $\zeta_w$ can be increased.

Two hard constraints (defined in Equation (3-3) and (3-4)) are imposed on the input space of the VSLs to reduce the size of the solution space and improve driver safety and comfort. The first constraint allows the VSLs to maximally change $\eta_t$ per sampling time. The second constraint allows the VSLs that are on two consecutive freeway segments to maximally differ $\eta_d$ from each other. Both constraints are chosen $\eta_t = \eta_d = 20$ km/h as in [23].

The weighting terms of the objective functions and parameters of the operational constraints are summarised in Table 4-5.

Table 4-5: Values of the objective function weights $\zeta_r$ and $\zeta_w$ for all the MPC algorithms and values for the operational constraint parameters $\eta_t$ and $\eta_d$ for the distributed and decentralised MPC algorithms.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\zeta_r$</td>
<td>0</td>
</tr>
<tr>
<td>$\zeta_w$</td>
<td>10</td>
</tr>
<tr>
<td>$\eta_t$</td>
<td>20 [km/h]</td>
</tr>
<tr>
<td>$\eta_d$</td>
<td>20 [km/h]</td>
</tr>
</tbody>
</table>

4-2-4 Number of iterations

The convergence of the system performance is investigated by increasing $n_{\text{dist}}$ and $n_{\text{alt}}$ to determine the optimal number of iterations for the proposed MPC algorithms. For proper investigation, the termination time of the algorithms is set to $t_{\text{term}} = \infty$. 

Master of Science Thesis

Uglješa Todorović
Number of alternating iterations

The convergence properties of the alternating optimisation are investigated by evaluating the system performance for an increasing \( n_{\text{alt}} \), whilst the number of distributed iterations is constant at \( n_{\text{dist}} = 1 \). The results are summarised in Table 4-6 and 4-7. The alternating optimisations of both algorithms converge within two iterations. Hence, \( n_{\text{alt}} = 2 \) is used for both algorithms in the rest of the work.

Table 4-6: This table shows the convergence behaviour of the alternating optimisation with FC-A-MPC by investigating the system performance for an increasing \( n_{\text{alt}} \). The alternating optimisation converges within two iterations.

<table>
<thead>
<tr>
<th>Controller</th>
<th>TTS [veh·h]</th>
<th>TTS_{\text{red}} [%]</th>
<th>( n_{\text{alt}} )</th>
<th>( n_{\text{dist}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>FC-A-MPC</td>
<td>4492</td>
<td>24.96</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>FC-A-MPC</td>
<td>4481</td>
<td>25.14</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>FC-A-MPC</td>
<td>4481</td>
<td>25.14</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>FC-A-MPC</td>
<td>4481</td>
<td>25.14</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4-7: This table shows the convergence behaviour of the alternating optimisation with DC-A-MPC by investigating the system performance for an increasing \( n_{\text{alt}} \). The alternating optimisation converges within two iterations.

<table>
<thead>
<tr>
<th>Controller</th>
<th>TTS [veh·h]</th>
<th>TTS_{\text{red}} [%]</th>
<th>( n_{\text{alt}} )</th>
<th>( n_{\text{dist}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC-A-MPC</td>
<td>4570</td>
<td>23.66</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DC-A-MPC</td>
<td>4516</td>
<td>24.56</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>DC-A-MPC</td>
<td>4516</td>
<td>24.56</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>DC-A-MPC</td>
<td>4516</td>
<td>24.56</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

Number of distributed iterations

The convergence properties of the distributed iterations are investigated by evaluating the system performance for an increasing \( n_{\text{dist}} \), whilst the number of alternating iterations is constant at \( n_{\text{alt}} = 2 \). The inputs that are found with the last distributed iteration are used as control inputs for the system, instead of the inputs of the iteration that yield the best objective function value as described in Chapter 3. This is done to properly investigate the convergence of the distributed iterations. The results are summarised in Table 4-8 and 4-9. The performance of both algorithms does not converge to the global optimum within four iterations. Hence, it makes sense to let the agents iterate for the whole controller sampling time (such that \( t_{\text{term}} = T_c \) and \( n_{\text{dist}} = \infty \)) and at the end use the inputs of the iteration that yield the best objective function values as described in Chapter 3.

Uglješa Todorović

Master of Science Thesis
The majority of MPC approaches that use VSLs as control measures relax the MINLP optimisation problems into nonlinear non-convex optimisation problems by considering the VSLs to be continuous decision variables instead of discrete decision variables [4, 5, 7, 25, 42, 43]. Since the VSLs are only allowed to take values from a prescribed discrete set, some approaches round the found VSLs to acceptable values. However, there are contradictory conclusions in the field on the effects of this relaxation, as some have found that it results in a significant performance loss in a centralised setting [23]. To extend these findings, this part of the case study investigates the effects of the relaxation on the system performance with a distributed control architecture.

To investigate these effects, the system performance of FC-A-MPC and DC-A-MPC is compared to the system performance of Fully Cooperative Rounding Model Predictive Control (FC-R-MPC) and Downstream Cooperative Rounding Model Predictive Control (DC-R-MPC). Both FC-R-MPC and DC-R-MPC have the same distributed architectures as the proposed algorithms, but consider the VSLs to be continuous decision variables in the optimisation and afterwards round the continuous VSLs to acceptable discrete values. The optimisation problem formulations and details on the optimisation algorithms are given in Appendix C.

Two specific cases are investigated: the first case only considers one distributed iteration $n_{\text{dist}} = 1$ and the second case considers multiple distributed iterations $n_{\text{dist}} = 4$. In both cases, the termination time is set to $t_{\text{term}} = \infty$, such that the convergence properties of the system performance for increasing $n_{\text{dist}}$ can be compared.
### 4-3-1 Results

Most of the supportive figures that describe the simulations can be found in Appendix D-1. However, the important results are summarised in Table 4-10 and 4-11. With the downstream cooperative architecture, the relaxation results in a performance loss of 3.77 % and 2.38 % for $n_{\text{dist}} = 1$ and $n_{\text{dist}} = 4$, respectively. With the fully cooperative architecture, the relaxation results in a performance loss of 4.42 % and 2.71 % for $n_{\text{dist}} = 1$ and $n_{\text{dist}} = 4$, respectively. Hence, in the worst case, the relaxation results in a decrease in $\text{TTS}_{\text{red}}$ of 3.31 % (corresponding to the fully cooperative architecture with one distributed iteration). Therefore, it can be concluded that the relaxation on the discrete nature of the VSLs results in a large performance with a distributed architecture. This confirms the results in [23] and extends them to the distributed case. Moreover, the performance loss is even more significant with $n_{\text{dist}} = 1$. Hence, the algorithms that use the alternating optimisation scheme find solutions that are much closer to the optimum with the first distributed iteration.

The speed limits $\text{VSL}_c$ that are found as continuous decision variables and their rounded solutions are shown in Figure 4-7 to 4-10 for the simulations with $n_{\text{dist}} = 4$. A large deviation between the continuous VSLs and the discretised solutions can be observed during uncongested of the simulations. These deviations result in the previously described losses in performance, as sub-optimal control inputs are used for the network.

<table>
<thead>
<tr>
<th>Controller</th>
<th>TTS [veh·h]</th>
<th>$\text{TTS}_{\text{red}}$ [%]</th>
<th>$n_{\text{dist}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC-A-MPC</td>
<td>4515</td>
<td>24.57</td>
<td>1</td>
</tr>
<tr>
<td>DC-A-MPC</td>
<td>4505</td>
<td>24.74</td>
<td>4</td>
</tr>
<tr>
<td>DC-R-MPC</td>
<td>4685</td>
<td>21.73</td>
<td>1</td>
</tr>
<tr>
<td>DC-R-MPC</td>
<td>4612</td>
<td>22.95</td>
<td>4</td>
</tr>
</tbody>
</table>

**Table 4-10:** The performance of considering the VSLs as discrete decision variables in the optimisation problems (DC-A-MPC) compared to considering them as continuous decision variables in the optimisation problems (DC-R-MPC) with a downstream cooperative architecture.

<table>
<thead>
<tr>
<th>Controller</th>
<th>TTS [veh·h]</th>
<th>$\text{TTS}_{\text{red}}$ [%]</th>
<th>$n_{\text{dist}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FC-A-MPC</td>
<td>4481</td>
<td>25.14</td>
<td>1</td>
</tr>
<tr>
<td>FC-A-MPC</td>
<td>4466</td>
<td>25.39</td>
<td>4</td>
</tr>
<tr>
<td>FC-R-MPC</td>
<td>4679</td>
<td>21.83</td>
<td>1</td>
</tr>
<tr>
<td>FC-R-MPC</td>
<td>4587</td>
<td>23.37</td>
<td>4</td>
</tr>
</tbody>
</table>

**Table 4-11:** The performance of considering the VSLs as discrete decision variables in the optimisation problems (FC-A-MPC) compared to considering them as continuous decision variables in the optimisation problems (FC-R-MPC) with a fully cooperative architecture.
Figure 4-7: The speed limits $VSL_c$ that are found as continuous decision variables and the rounded solutions with DC-R-MPC and $n_{dist} = 4$.

Figure 4-8: The speed limits $VSL_c$ that are found as continuous decision variables and the rounded solutions with DC-R-MPC and $n_{dist} = 4$.

Figure 4-9: The speed limits $VSL_c$ that are found as continuous decision variables and the rounded solutions with FC-R-MPC and $n_{dist} = 4$. 
4-3-2 Conclusions

The effects of relaxing the MINLP optimisation problems into non-convex nonlinear continuous optimisation problems by considering the VSLs as continuous decision variables have been investigated with a distributed control architecture. In the worst case (with a fully cooperative architecture and one distributed iteration), the relaxation results in a performance loss of 4.42%. Hence, the relaxation results in a significant performance loss with a distributed control architecture.

4-4 Case study B: Architecture comparison

In this part of the case study, the proposed distributed algorithms are compared to the more conventional centralised and decentralised algorithms in terms of system performance and computational complexity. Both the centralised and decentralised algorithms use an alternating optimisation scheme to solve the MINLP optimisation problems. Hence, the algorithms are denoted as Centralised Alternating Model Predictive Control (Cent-A-MPC) and Decentralised Alternating Model Predictive Control (Dec-A-MPC).

Most of the control parameters that are determined in Section 4-2 for the proposed distributed algorithms can be identically used for the Cent-A-MPC and Dec-A-MPC. More specifically, the horizons, controller sampling times and objective functions weights are chosen the same, such that \(N_u = 3\), \(N_p = 10\), \(T_c = 120\) s, \(\zeta_r = 0\) and \(\zeta_w = 10\) are used for the decentralised and centralised algorithms as well. Moreover, Dec-A-MPC requires the same number of alternating iterations \(n_{alt} = 2\) as the distributed algorithms to converge and also uses \(\eta_L = \eta_{kl} = 20\) km/h. The solution space of the centralised algorithm is much larger. Therefore, Cent-A-MPC requires \(n_{alt} = 5\) for good system performance. The optimisation problem formulation and details on the optimisation algorithms are given in Appendix C.

4-4-1 Results

Firstly, the performance of the centralised, decentralised and both distributed MPC algorithms is discussed individually by looking at the determined inputs and the states of the...
traffic network. Then, at the end of this section, the different algorithms are compared with each other by looking at system performance, computational complexity and the traffic behaviour at the bottleneck locations of the network.

**Cent-A-MPC**

Figure 4-11 to 4-14 show the simulation results when the centralised algorithm Cent-A-MPC controls the network. The algorithm results in a system performance of $\text{TTS} = 4452 \text{ veh} \cdot \text{h}$ (corresponding to $\text{TTS}_{\text{rel}} = 25.63\%$) and has a computational complexity of $\text{CT}_{\text{max}} = 10999 \text{ s}$. One agent determines the control inputs for the whole network by using one comprehensive system model. Hence, the centralised algorithm has the optimal system performance, since the control measures are optimally coordinated. The network remains mostly uncongested, as can be seen in Figure 4-11. However, because one comprehensive system model is used, the computational complexity is exceptionally high. The agent does not manage to solve the optimisation problems within the controller sampling intervals and, therefore, the algorithm is not implementable in real-time. There is one critical point in the simulation when all three on-ramp queues are close to the soft constraint limit of $w_{\text{max}} = 100 \text{ veh}$. At this point, the agent decides that slightly violating the soft constraint at the second on-ramp is beneficial for the system performance. At this critical point, all VSLs drop to 40 km/h to dissolve the queues. The queue length at the second on-ramp remains within 10% of the soft constraint limit. A heat map of the average speeds in the network is given in Figure D-41 and the CT for every controller sample is given in Figure D-42 in Appendix D-2.

![Figure 4-11: A heat map of the densities of all the segments when the Cent-A-MPC algorithm controls the network.](image-url)
Figure 4-12: The queue lengths in front of the three on-ramps when the Cent-A-MPC algorithm controls the network.

Figure 4-13: The metering rates of the three RM installations when the Cent-A-MPC algorithm controls the network.

Figure 4-14: The values of the six VSLs when the Cent-A-MPC algorithm controls the network.
Dec-A-MPC

Figure 4-15 to 4-18 show the simulation results when the decentralised algorithm Dec-A-MPC controls the network. The algorithm results in a system performance of $TTS = 4738$ veh-h (corresponding to $TTS_{red} = 20.85\%$) and has a computational complexity of $CT_{\text{max}} = 5$ s. The algorithm partitions the network in three subsystems controlled by three agents. The computational complexity is low because the agents do not cooperate with each other. Hence, all agents manage to solve their optimisation problems within the controller sampling intervals. However, the system performance is also rather sub-optimal because the control inputs are not coordinated due to the lack of cooperation between agents. The first two agents keep their on-ramps open during the whole simulation because their respective subsystems remain uncongested (illustrated in Figure 4-17). Because of this, a significant traffic jam originates in front of the third on-ramp (illustrated in Figure 4-15). Consequently, vehicles in the queue cannot enter the mainstream sufficiently fast and the soft constraint of $w_{\text{max}} = 100$ veh is violated significantly at the third on-ramp (illustrated in Figure 4-16). A heat map of the average speeds in the network is given in Figure D-43 in Appendix D-2.

Figure 4-15: A heat map of the densities of all the segments when the Dec-A-MPC algorithm controls the network.

Figure 4-16: The queue lengths in front of the three on-ramps when the Dec-A-MPC algorithm controls the network.
Figure 4-17: The metering rates of the three RM installations when the Dec-A-MPC algorithm controls the network.

Figure 4-18: The values of the six VSLs when the Dec-A-MPC algorithm controls the network.

**FC-A-MPC**

Figure 4-19 to 4-22 show the simulation results when the fully cooperative algorithm FC-A-MPC controls the network. The algorithm results in a system performance of TTS = 4463 veh·h (corresponding to TTS$_{red}$ = 25.44 %) and has a computational complexity of CT$_{max}$ = 120 s. The computational complexity is relatively low because the network is partitioned into three subsystems controlled by three agents. As a result, all agents manage to solve their optimisation problems within the controller sampling intervals and the algorithm is implementable in real-time. Moreover, the network remains mostly uncongested (illustrated in Figure 4-19) because the agents control their respective part of the network, whilst considering the global network performance. There is one critical point in the simulation when all three on-ramp queues are close to the soft constraint limit of $w_{max}$ = 100 veh. At this point, the agents decide that slightly violating the soft constraint at the second on-ramp is beneficial for the global system performance. At this critical point, all six VSLs drop to 40 km/h to dissolve the queues. The queue length at the second on-ramp remains within 10 % of the soft constraint. A heat map of the average speeds in the network is given in Figure D-44 in Appendix D-2.
Figure 4-19: A heat map of the densities of all the segments when the FC-A-MPC algorithm controls the network.

Figure 4-20: The queue lengths in front of the three on-ramps when the FC-A-MPC algorithm controls the network.

Figure 4-21: The metering rates of the three RM installations when the FC-A-MPC algorithm controls the network.
Figure 4-22: The values of the six VSLs when the FC-A-MPC algorithm controls the network.

**DC-A-MPC**

Figure 4-23 to 4-26 show the simulation results when the downstream cooperative algorithm DC-A-MPC controls the network. The algorithm results in a system performance of \( \text{TTS} = 4505 \text{ veh-h} \) (corresponding to \( \text{TTS}_{\text{red}} = 24.74 \% \)) and has a computational complexity of \( \text{CT}_{\text{max}} = 120 \text{ s} \). The computational complexity is relatively low because the network is partitioned into three subsystems controlled by three agents. As a result, all agents manage to solve the optimisation problems within the controller sampling intervals and the algorithm is implementable in real-time. Although the system performance is relatively good, the agents do not manage to keep the network uncongested (illustrated in Figure 4-23). The first on-ramp is open for a large part of the simulation (illustrated in Figure 4-25) because the agents are only partly cooperating. As a consequence, a traffic jam occurs in front of the third on-ramp. A heat map of the average speeds in the network is given in Figure D-45 in Appendix D-2.

Figure 4-23: A heat map of the densities of all the segments when the DC-A-MPC algorithm controls the network.
Figure 4-24: The queue lengths in front of the three on-ramps when the DC-A-MPC algorithm controls the network.

Figure 4-25: The metering rates of the three RM installations when the DC-A-MPC algorithm controls the network.

Figure 4-26: The values of the six VSLs when the DC-A-MPC algorithm controls the network.
4-4-2 Comparison

Performance comparison

The performance of the four algorithms and the no-control case is summarised in Table 4-12. As expected, the centralised algorithm achieves the best system performance. However, because only one agent controls the network by using one comprehensive system model, the computational complexity is exceptionally high. Hence, the algorithm is far from implementable in real-time because the optimisation problems cannot be solved within the controller sampling intervals. On the other hand, the decentralised algorithm has, as expected, the lowest computational complexity because three agents are controlling the network without cooperation. However, because the agents do not cooperate, the system performance is rather sub-optimal and the soft constraint on the queue length at the third on-ramp is violated significantly.

Both proposed distributed algorithms offer a good trade-off between computational complexity and system performance. FC-A-MPC achieves a system performance that is similar to the optimal performance of the centralised algorithm and, hence, the agents manage to keep the network mostly uncongested. DC-A-MPC results in a slightly worse system performance than FC-A-MPC because the agents are not fully cooperating. As a consequence, a traffic jam originates in front of the third on-ramp. However, both algorithms have a significantly better system performance than the decentralised algorithm. Moreover, the computational complexity of the proposed algorithms is significantly lower than the computational complexity of the centralised algorithm because the network is controlled by three agents. As a result, the agents manage to solve the optimisation problems within the controller sampling intervals and, hence, both algorithms are implementable in real-time. Furthermore, DC-A-MPC has a lower computational complexity than FC-A-MPC. Where the computational complexity of FC-A-MPC still scales with the size of the network, the computational complexity of DC-A-MPC only scales with the size of the subsystems.


<table>
<thead>
<tr>
<th>Controller</th>
<th>TTS [veh·h]</th>
<th>TTSred [%]</th>
<th>CTmax [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>No control</td>
<td>5986</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Dec-A-MPC</td>
<td>4738</td>
<td>20.85</td>
<td>5</td>
</tr>
<tr>
<td>DC-A-MPC</td>
<td>4505</td>
<td>24.74</td>
<td>120</td>
</tr>
<tr>
<td>FC-A-MPC</td>
<td>4463</td>
<td>25.44</td>
<td>120</td>
</tr>
<tr>
<td>Cent-A-MPC</td>
<td>4452</td>
<td>25.63</td>
<td>10999</td>
</tr>
</tbody>
</table>

Bottleneck comparison

There are three bottlenecks in the network, located at the segments that contain on-ramps: segments 7, 14 and 21. The flows and densities of those segments for all algorithms and the no-control case are shown in Figure 4-27 to 4-32. The densities of all three bottlenecks exceed the critical density when no control is applied. On the contrary, all control algorithms manage to keep the densities of the first two bottlenecks below the critical density. Moreover,
none of the control algorithms manages to keep the density at the last bottleneck below the
critical density. However, the densities on this bottleneck with both Cent-A-MPC and FC-A-
MPC are very similar and remain much closer to the critical density than the densities with
Dec-A-MPC and DC-A-MPC.

**Figure 4-27:** A comparison of the densities at the first bottleneck located at segment 7 for all
controllers and the no-control case.

**Figure 4-28:** A comparison of the flows at the first bottleneck located at segment 7 for all
controllers and the no-control case.

**Figure 4-29:** A comparison of the densities at the second bottleneck located at segment 14 for
all controllers and the no-control case.
Figure 4-30: A comparison of the flows at the second bottleneck located at segment 14 for all controllers and the no-control case.

Figure 4-31: A comparison of the densities at the third bottleneck located at segment 21 for all controllers and the no-control case.

Figure 4-32: A comparison of the flows at the third bottleneck located at segment 21 for all controllers and the no-control case.
4-4-3 Conclusions

The performance of Cent-A-MPC, Dec-A-MPC, FC-A-MPC and DC-A-MPC has been compared on a hypothetical freeway network. The centralised algorithm results in the best system performance, but also the highest computational complexity. The decentralised algorithm results in the worst system performance, but also the lowest computational complexity. Both FC-A-MPC and DC-A-MPC offer a good trade-off between computational complexity and system performance. Where FC-A-MPC has a system performance that is similar to the optimal solution of the centralised algorithm, DC-A-MPC has a lower computational complexity, which only scales with the size of the subsystems. Dec-A-MPC, FC-A-MPC and DC-A-MPC are implementable in real-time, since they manage to solve their optimisation problems within the controller sampling intervals. On the contrary, Cent-A-MPC does not manage to solve the optimisation problems within the controller sampling intervals.

4-5 Conclusions

In this chapter, the proposed algorithms, FC-A-MPC and DC-A-MPC, have been evaluated. This has been done by comparing their performance to the performance of more conventional MPC algorithms in a case study.

Firstly, the effects on the system performance by relaxing the VSLs to be continuous decision variables have been investigated. In the worst case (with a fully cooperative architecture and one distributed iteration), the relaxation results in a performance loss of 4.42%. Hence, it can be concluded that the relaxation results in a significant performance loss with a distributed architecture. This confirms the results in [23] and extends them to the distributed case.

Secondly, the performance of the proposed distributed algorithms has been compared to the performance of the more conventional decentralised and centralised algorithms in terms of system performance and computational complexity. As expected, the centralised algorithm has the best system performance ($TTS_{\text{red}} = 25.63\%$). However, it also has the highest computational complexity ($CT_{\text{max}} = 10999$ s) and therefore it is not implementable in real-time. On the other hand, the decentralised algorithm has the worst system performance ($TTS_{\text{red}} = 20.85\%$) but also the lowest computational complexity ($CT_{\text{max}} = 5$ s). Both FC-A-MPC and DC-A-MPC offer a good trade-off between computational complexity and system performance. FC-A-MPC has a better system performance than DC-A-MPC ($TTS_{\text{red}} = 25.44\%$ and $TTS_{\text{red}} = 24.74\%$, respectively), which is similar to the optimal solution of the centralised algorithm. Moreover, both algorithms drastically reduce the computational complexity of the problem by combining a distributed control architecture with an alternating optimisation scheme. Hence, the agents manage to solve all optimisation problems within the controller sampling intervals of 120 s and, thus, both algorithms are implementable in real-time. Furthermore, DC-A-MPC is less complex from a computational point of view, as the complexity of the algorithm only scales with the size of the subsystems.

For this freeway network, FC-A-MPC has the best overall performance, as it has the best system performance of the algorithms that manage to solve the optimisation problems within the controller sampling intervals. However, for larger freeway networks, the algorithm might still be too complex from a computational point of view. For such freeway networks, DC-A-MPC could still be implementable.
This chapter concludes the thesis and suggests relevant areas for future research. Section 5-1 draws the main conclusions by answering the research questions and stating the contributions to the state-of-the-art. Finally, Section 5-2 suggests relevant areas for future research.

5-1 Conclusions

5-1-1 Research questions

The goal of this thesis was to answer the following main research question:

*Is it possible to develop distributed MPC algorithms for large freeway networks using discrete variable speed limits and continuous ramp metering rates that offer a good trade-off between computational complexity and system performance?*

Two sub-questions were derived to answer the main question:

1. *Does relaxing the variable speed limits to be continuous decision variables in the optimisation result in a significant performance loss with a distributed architecture?*

2. *How is the performance of the proposed distributed algorithms compared to centralised and decentralised algorithms in terms of system performance and computational complexity?*

Two distributed MPC algorithms that use an alternating optimisation scheme, FC-A-MPC and DC-A-MPC, have been developed and their performance has been evaluated in a case study to answer these questions.
First sub-question

Most related works that use VSLs as control measures relax these to be continuous decision variables instead of discrete decision variables in the optimisation problems [4, 5, 7, 25, 42, 43]. This relaxation transforms the initial MINLP optimisation problems into nonlinear non-convex continuous optimisation problems, which are substantially easier to solve. However, there are contradictory conclusions in the field on the effects of this relaxation on the system performance, as some have found that it results in a significant performance loss with a centralised architecture [23].

To answer this open question in the field, this thesis has investigated the effects of this relaxation with a distributed architecture. The proposed algorithms have been compared to two similar algorithms that have the same distributed architectures but consider the VSLs as continuous decision variables in the optimisation.

The largest performance loss in the case study occurs with a fully cooperative architecture and one distributed iteration. In this case, the algorithm that considers the VSLs as discrete decision variables results in a reduction in TTS of 25.14 % compared to the uncontrolled network, whilst the algorithm that relaxes the VSLs to be continuous decision variables results in a reduction in TTS of 21.83 %. Hence, it can be concluded that the relaxation results in a significant performance loss with a distributed architecture because sub-optimal inputs are used for the network. Therefore, the VSLs should be discrete decision variables in the optimisation for good performance.

Second sub-question

The performance of the proposed algorithms has been compared to the performance of the more conventional centralised and decentralised algorithms. Their performance serves as a benchmark, as the centralised algorithm generally has the optimal system performance in a receding horizon context, whilst the decentralised algorithm generally is the least complex from a computational viewpoint.

The centralised algorithm has the best system performance in the case study, resulting in a reduction in TTS of 25.63 % relative to the uncontrolled network. However, it also has an exceptionally high computational complexity, peaking at a computational time of 10999 s for one controller sample. Hence, the centralised algorithm is far from implementable in real-time because it does not manage to solve the optimisation problems within the controller sampling intervals of 120 s.

On the other hand, the decentralised algorithm has the worst system performance with a reduction in TTS of 20.85 %. However, it also has the lowest computational complexity, peaking at a computational time of 5 s for one controller sample. Hence, the decentralised algorithm is implementable in real-time, but results in a rather sub-optimal system performance.

Both proposed algorithms offer a good trade-off between computational complexity and system performance. FC-A-MPC results in a reduction in TTS of 25.44 %, which is remarkably similar to the optimal performance of the centralised algorithm. DC-A-MPC has a slightly worse system performance with a reduction in TTS of 24.74 %. Moreover, both algorithms sufficiently reduce the computational complexity of the problem, such that all optimisation
problems are solved within the controller sampling intervals. Hence, both proposed algorithms are implementable in real-time. Furthermore, DC-A-MPC is less complex than FC-A-MPC from a computational viewpoint, as the computational complexity of the algorithm only scales with the sizes of the subsystems instead of the size of the network.

**Main research question**

This work has shown that combining a distributed control architecture with an alternating optimisation scheme is an outstanding approach to develop MPC algorithms that offer a good trade-off between computational complexity and system performance. The approach reduces the computational complexity of the problem, but still ensures a good system performance by considering the VSLs as discrete decision variables with the alternating optimisation scheme and coordinating the control measures with the distributed control architecture.

Two distributed MPC algorithms have been developed that both offer a different trade-off between computational complexity and system performance. FC-A-MPC outperforms DC-A-MPC with a system performance that is remarkably similar to the optimal performance of centralised MPC. DC-A-MPC outperforms FC-A-MPC with a computational complexity that only scales with the size of the subsystems.

It highly depends on the size of the freeway network which algorithm is more suitable. For the hypothetical freeway network in the case study, FC-A-MPC has the best overall performance, as it has the best system performance of the algorithms that manage to solve the optimisation problems within the controller sampling intervals. However, for even larger freeway networks, FC-A-MPC might computationally become too complex, whilst DC-A-MPC could still be implementable.

**5-1-2 Contributions**

This thesis has two main contributions to the state-of-the-art:

1. The main contribution of this thesis is the proposal of two novel MPC algorithms for coordinated control of RM installations and VSLs. Both algorithms use a distributed control architecture and an alternating optimisation scheme to offer a trade-off between computational complexity of system performance.

2. This thesis has shown that relaxing the VSLs to be continuous decision variables instead of discrete decision variables in the optimisation results in a significant performance loss with a distributed architecture because sub-optimal inputs are used for the network. This result confirms the findings in [23] and extends them to the distributed case. Hence, the contradictory conclusions in the field on the effects of this relaxation are disproven more strongly now.

A paper, which can be found in Appendix E, has been written on these contributions.
5-2 Future work

Even though the proposed algorithms have shown great results for coordinated control of VSLs and RM installations, several challenges and opportunities for future research remain. Some recommendations are outlined in the remainder of this section.

Include prediction of future demands

The future demands are required in order to determine future control inputs in the MPC framework. Most relevant works assume these future demands to be known. However, the proposed algorithms use the measurements of the demands as constant demands over the prediction horizon, since assuming the demands to be known is less realistic. Hence, the performance of the proposed algorithms can be improved by including an estimation of the future demands. The estimation of traffic demands with CARMA, CARIMA and exponential estimation has been investigated in [22].

Investigate robustness w.r.t. measurements errors

This work uses perfect measurements of the states and demands in order to determine the optimal control inputs in the case study. However, in real-life implementations, the measurements contain errors. Hence, the robustness of the proposed algorithms needs to be investigated w.r.t. those measurement errors. Robust MPC approaches with application to freeway traffic control have been proposed in [46].

Communication delays and errors

The case study assumes that there is no delay in the communication of measurements of the network and decision variables between agents. Moreover, it assumes that no errors occur with the communication. However, in real-life implementations, there are always delays in communication and errors occur like missed samples. Hence, the algorithms have to be made robust against these types of delays and errors.

Evaluate algorithms in real-life

The case study uses a hypothetical freeway network to evaluate the performance of the algorithms. For this reason, the parameters of METANET, describing the prediction model and the simulation model, are taken from literature and, subsequently, no system identification is necessary. Therefore, there is no mismatch between the prediction model and simulation model, i.e. the prediction model is perfectly accurate. However, it would be interesting to evaluate the performance of the algorithms on a real-life freeway. For implementation, the parameters of METANET would have to be estimated, resulting in an inaccuracy of the prediction model. Investigation of the effects of this model mismatch on the performance of the algorithms would be interesting. A parameter identification algorithm for METANET has been proposed in [27]. Moreover, the effects of the previously described phenomena (measurement errors, communication delays and errors) could be investigated as well.
Other objective functions

The objective functions include the TTS as the main performance criterion. However, it would be interesting to investigate the performance of the proposed algorithms with other performance criteria in the objective functions such as total emission [36, 44] and total fuel consumption [36, 44].

Integrating other control measures

The algorithms can be extended with the control of other control measures such as reversible lanes [29] and lane change [55]. It would be interesting to investigate the influence on the performance by adding those control measures.

Combining the MLD model of METANET with a distributed architecture

The combination of discrete VSLs and continuous RM rates results in an MINLP optimisation problem that every agent has to solve. This work relaxes the MINLP problems with an alternating optimisation scheme. However, another approach is to rewrite METANET as an MLD model [36] and relax the MINLP problems into MILP problems. It would be interesting to compare the performance of both approaches with a distributed architecture.
Appendix A

Model Predictive Control

The term MPC does not designate a specific control strategy, but rather a control framework. All methods in this framework use the following main concepts [17]:

- **Prediction model**: a model of the system is used to predict relevant future system trajectories.

- **Objective function**: an objective function is minimised to find an optimal sequence of control inputs. Constraints can be imposed on the states and inputs with the optimisation.

- **Receding horizon strategy**: only the input that is found for the first controller time step is applied to the system. The input sequence that is found for the rest of the horizon is discarded. This process is iterated for the next controller time steps.

![Figure A-1: An illustration of the receding horizon strategy that is used with MPC.](image)

Master of Science Thesis

Uglješa Todorović
Hence, the various MPC methods differ mainly in prediction models, objective functions, optimisation methods and control architectures that are used. The methods in the MPC framework use the following strategy [17]:

1. The future outputs $y(k + l|k)$ for $l = 1, ..., N_p$ are predicted for all samples in the prediction horizon $N_p$ at discrete time instant $k$. These outputs depend on the past inputs and outputs and on the future control signal $u(k + l|k)$ for $l = 0, ..., N_p - 1$.

2. The set of future control inputs are calculated by optimising an objective function, which often attempts to keep the process as close as possible to a reference trajectory.

3. Only control input $u(k|k)$ is sent to the system and all the other input samples are discarded. This process is repeated for all next controller time steps.

The strategy is illustrated in Figure A-1.
Appendix B

METANET

Introduction

METANET is a deterministic model that can simulate traffic flow in freeway networks of arbitrary topology including freeway stretches, bifurcations, on-ramps and off-ramps [35]. It is a second-order macroscopic model, discrete in both space and time and models traffic flow analogous with a compressible fluid. Furthermore, it is capable of modelling control measures such as VSLs, RM installations and route guidance.

The freeway is partitioned into links, similar to a directed graph. The links represent homogeneous freeway stretches and are interconnected by nodes. Consequently, major changes in geometry are modelled by adding a new link.

The links are split into segments. The state of each segment is described by three aggregated variables: the vehicular density $\rho$ (in veh/km/lane), the vehicular flow $q$ (in veh/h) and the space-mean speed $v$ (in km/h). The computational speed of METANET is not affected by vehicular density due to these aggregated variables. Hence, METANET is highly suitable for model-based control.

The remainder of this section describes the model equations of METANET and the boundary conditions that are used in this work.

Model equations

Each link $m$ is divided into $N_m$ segments $i$ of respective length $L_{m,i}$. An example of a link in METANET is depicted in Figure B-1. However, this work only differentiates between segments to improve readability. This is possible because the network in the case study in Chapter 4 can be modelled by one large link.

With the segmentation, the following constraint has to be satisfied:

$$L_i > v_{\text{free}} T, \quad (B-1)$$
where $v_{\text{free}}$ is the average speed of drivers in free-flow conditions and $T$ is the sampling time. The constraint is necessary for computational stability.

The sampling time $T$ is related to time $t$ and sample counter $k$ by:

$$t = kT.$$  \hfill (B-2)

The outflow $q_i(k)$ of segment $i$ during period $[kT, (k+1)T]$ is an auxiliary variable and is described by:

$$q_i(k) = \lambda v_i(k) \rho_i(k),$$ \hfill (B-3)

where $\lambda$ is the number of lanes, $v_i(k)$ and $\rho_i(k)$ are the space-mean speed and the vehicular density of segment $i$ at time $k$.

The update equation of $\rho_i(k)$ is derived from the conservation of vehicles, analogous with hydrodynamic particle flow:

$$\rho_i(k+1) = \rho_i(k) + \frac{T}{\lambda L_i} \left( 1 - \beta_i(k) \right) q_{i-1}(k) - q_i(k) + q_{r,i}(k),$$ \hfill (B-4)

where $\beta_i(k)$ is the fraction of flow that leaves segment $i$ from an off-ramp and $q_{r,i}(k)$ is the flow that enters segment $i$ from and on-ramp. Hence, if segment $i$ does not contain an on-ramp or off-ramp, $\beta_i(k)$ and $q_{r,i}(k)$ are, respectively, zero. The equation implies that the number of vehicles in segment $i$ increases according to the difference in inflow and outflow at the boundaries of the segment.

The speed dynamics are described in a heuristic way. The update equation for the space-mean speed $v_i(k)$ of segment $i$ at time $k$ involves a relaxation term that describes the drivers will to achieve a desired speed $V(\rho_i(k))$, a convection term that describes the change in speed that is caused by the inflow of upstream vehicles and an anticipation term that expresses the change in speed of a segment that is associated with the difference in density level with a downstream segment:

$$v_i(k+1) = v_i(k) + \frac{T}{\tau L_i} \left[ V(\rho_i(k)) - v_i(k) \right] +$$
$$\frac{T}{L_i} v_i(k) \left[ v_{i-1}(k) - v_i(k) \right] -$$
$$\frac{\eta T}{\tau L_i} \frac{[\rho_{i+1}(k) - \rho_i(k)]}{\rho_i(k) + \kappa},$$ \hfill (B-5)
where \( \tau, \eta \) and \( \kappa \) are model parameters, and the desired speed \( V(\rho_i(k)) \) is described by:

\[
V(\rho_i(k)) = v_{\text{free}} \exp \left[ -\frac{1}{a} \left( \frac{\rho_i(k)}{\rho_{\text{crit}}} \right)^a \right],
\]

(B-6)

where \( a \) is a model parameter and \( \rho_{\text{crit}} \) is the critical density, which is the density that corresponds to the maximum vehicular flow. Equation (B-6) describes a fundamental diagram, depicted in Figure 1-1.

The VSLs are modelled according to [4] by modifying Equation (B-6) for the segments that contain an on-ramp to:

\[
V(\rho_i(k)) = \min \left[ v_{\text{free}} \exp \left[ -\frac{1}{a} \left( \frac{\rho_i(k)}{\rho_{\text{crit}}} \right)^a \right], (1 + \alpha) \cdot U_{\text{VSL}}^i(k) \right],
\]

(B-7)

where \( U_{\text{VSL}}^i(k) \) is the variable speed limit imposed on segment \( i \) at time \( k \) and \( \alpha \) is a non-compliance factor of drivers. Other approaches for modelling VSLs are presented in [28].

The change in length of the queue \( w_i(k) \) at the on-ramp of segment \( i \) is described by:

\[
w_i(k + 1) = w_i(k) + T \left[ d_i(k) - q_{r,i}(k) \right],
\]

(B-8)

where \( d_i(k) \) is the demand at the on-ramp of segment \( i \).

The outflow \( q_{r,i}(k) \) depends on the available traffic at time \( k \), the maximum flow allowed by the metering rate and the maximum flow that can enter the freeway due to mainstream conditions:

\[
q_{r,i}(k) = \min \left[ d_i(k) + \frac{w_i(k)}{T}, C \cdot U_i^r(k), C \cdot \left( \frac{\rho_{\text{max}} - \rho_i(k)}{\rho_{\text{max}} - \rho_{\text{crit}}} \right) \right],
\]

(B-9)

where \( C \) is the on-ramp capacity, \( U_i^r(k) \in [0, 1] \) is the metering rate and \( \rho_{\text{max}} \) represents the maximum density of a segment under jammed conditions.

If a segment contains an on-ramp, the speed drop caused due to merging phenomena is modelled by adding

\[
- \frac{\delta T q_{r,i}(k) v_i(k)}{L_i \lambda [\rho_i(k) + \kappa]}
\]

(B-10)

to Equation (B-5), where \( \delta \) is a model parameter.

Generally, drivers tend to decelerate faster than they tend to accelerate. For this reason, the anticipation term \( \eta \) in Equation (B-5) takes different values for when the downstream density is higher or lower than the actual segment [4]:

\[
\eta = \begin{cases} 
\eta_h & \text{if } \rho_{i+1}(k) \geq \rho_i(k) \\
\eta_l & \text{otherwise}
\end{cases},
\]

(B-11)

where \( \eta_h \) and \( \eta_l \) are anticipation terms for the two different cases.
Boundary conditions

The traffic state of a segment $i$ depends on the upstream speed $v_{i-1}(k)$, the upstream flow $q_{i-1}(k)$ and the downstream density $\rho_{i+1}(k)$. Therefore, the upstream flow and upstream speed of the mainstream origin $o$ of the network and the downstream density of the downstream exit of the network need to be described with boundary conditions.

A frequently used approach [25] is to assume that the inflow $q_o$ at the mainstream origin $o$ is limited by the maximum flow $q_{\text{max}}$, and follows from the speed-flow relationship in Equation (B-3) and (B-6):

$$q_o(k) = \min \left[ d_o(k) + \frac{w_o(k)}{T}, q_{\text{max}}(k) \right],$$  \hspace{1cm} (B-12)

where $q_{\text{max}}(k)$ is given by:

$$q_{\text{max}}(k) = \begin{cases} \lambda v_o(k) \rho_{\text{crit}} \left[ -a \ln \left( \frac{v_o(k)}{v_{\text{free}}} \right) \right]^\frac{1}{2} & \text{if } v_o(k) < V(\rho_{\text{crit}}) \\ \lambda V(\rho_{\text{crit}}) \rho_{\text{crit}} & \text{otherwise} \end{cases},$$  \hspace{1cm} (B-13)

and the vehicular speed $v_o(k)$ at the mainstream origin $o$ is given by:

$$v_o(k) = v_1(k).$$  \hspace{1cm} (B-14)

Finally, the downstream density at the exit of the mean-stream is defined as [25]:

$$\rho_{N_{\text{all}}+1}(k) = \min \left[ \rho_{N_{\text{all}}}(k), \rho_{\text{crit}} \right].$$  \hspace{1cm} (B-15)
Appendix C

Optimisation Problem Details

This appendix formalises the optimisation problems of the algorithms that are used in the case study but are not formalised in the core of the thesis. Moreover, this appendix provides details on all the optimisation methods that are used.

C-1 FC-A-MPC and DC-A-MPC

FC-A-MPC and DC-A-MPC use an alternating optimisation scheme to relax the MINLP optimisation problems into two sub-problems: nonlinear non-convex continuous optimisation problems for finding RM rates (defined in Equation (3-5) and (3-11) for FC-A-MPC and DC-A-MPC, respectively) and discrete optimisation problems for finding VSLs (defined in Equation (3-8) and (3-14) for FC-A-MPC and DC-A-MPC, respectively). The alternating optimisation schemes are described in Section 3-2-4. In the remainder of this section, the optimisation methods that are used to solve both sub-problems are discussed.

C-1-1 Discrete optimisation algorithm details

The discrete optimisation problems are solved by evaluating all possible solutions of the search tree. Hence, the objective function values for all the feasible VSL combinations are evaluated and the combination that yields the lowest value is the optimal solution.

The set of all feasible VSL input combinations is generated by permuting all acceptable VSL values in VSL\textsubscript{set} over the prediction horizon \(N_p\) for both VSLs in the subsystem. The size of this set is reduced by the control horizon \(N_u\) and operational constraints in Equation (3-3) and (3-4). The size of this set, and therefore the computational time that is needed to solve the optimisation problem, grows exponentially with the number of VSLs in the subsystem, \(N_u\), and the size of VSL\textsubscript{set}. However, since only two VSLs per subsystem are considered and both \(N_u\) and VSL\textsubscript{set} are relatively small, the problem can be solved sufficiently fast. Moreover, since all possible combinations are evaluated, this method yields the global optimum to the discrete optimisation problem.
C-1-2 Continuous optimisation algorithm details

The nonlinear non-convex continuous optimisation problems are solved by combining SQP with a multi-start approach. With a multi-start approach, multiple optimisation problems are evaluated with different initial points. By increasing the number of initial points, it is more likely that the global optimum is attained although it cannot be guaranteed. However, as the solution space of the problem is small since only one RM installation is considered per agent, the number of initial points that have to be evaluated is also small. All possible permutations of the values 0.1, 0.25, 0.45 and 0.8, constant over the horizon, are used as initial point profiles. Additionally, two initial point profiles are used, that linearly increase and linearly decrease over the horizon between the values 0.1 and 0.8. Hence, a total of 6 initial point profiles are evaluated with the multi-start approach.

The \textit{fmincon} function of the Optimization Toolbox of MATLAB is used to solve the continuous optimisation problems with SQP. The default SQP optimisation options are used, with some modifications:

- The setting \textit{MaxFunctionEvaluations} is set to $10 \cdot 10^6$,
- The setting \textit{MaxIterations} is set to $10 \cdot 10^6$.

C-2 FC-R-MPC and DC-R-MPC

Both FC-R-MPC and DC-R-MPC have the same distributed architectures as FC-A-MPC and DC-A-MPC, but do not use an alternating optimisation scheme to solve the MINLP problems. Instead, FC-R-MPC and DC-R-MPC relax the MINLP optimisation problems into nonlinear non-convex continuous optimisation problems by considering the VSLs to be continuous decision variables. The agents solve their optimisation problems in a distributed manner and communicate intermediate solutions to neighbours. After $n_{\text{dist}}$ iterations, the found VSLs are rounded to acceptable values, and subsequently, the global objective function values are evaluated with the found inputs in every iteration. The first time samples of the inputs that yield the lowest global objective function value are sent to the network.

In the remainder of this section, the optimisation problems for FC-R-MPC and DC-R-MPC are formulated, and the method that is used to solve the continuous optimisation problems is described.

C-2-1 Problem formulation

The optimisation problems that the agents have to solve with both FC-R-MPC and DC-R-MPC are very similar. The only differences between the optimisation problems are the objective functions, and the way that $\hat{U}_r^s$, $\hat{U}_{\text{VSL}}^s$, $x$ and $d$ are defined.

The optimisation problem that agent $s$ has to solve with both algorithms is defined as:
\[
\begin{align*}
\min_{U_t^e(k_c), U_{VSL}^s(k_c)} & \quad J^s(k_c) \\
\text{subject to:} \quad x(k_c + l + 1) &= f_M((x(k_c + l), \bar{U}_r^s(k_c + l), \bar{U}_{VSL}^s(k_c + l), \\
&\quad \hat{U}_r^s(k_c + l), \hat{U}_{VSL}^s(k_c + l), d(k_c + l)), \\
x(k_c) &= x_k, \\
d(k_c + l) &= d_k, \\
U_t^i(k_c + l) &\in \mathcal{U}_t, \forall i \in I_t^s, \\
U_{VSL}^z(k_c + l) &\in \mathcal{U}_{VSL}, \forall z \in I_{VSL}^s, \\
U_t^i(k_c + j) &= U_t^i(k_c + N_c - 1), \forall i \in I_t^s, \\
U_{VSL}^z(k_c + j) &= U_{VSL}^z(k_c + N_c - 1), \forall z \in I_{VSL}^s, \\
\text{for } l &\in \{0, 1, \ldots, N_p - 1\}, \\
\text{for } j &\in \{N_u, \ldots, N_p - 1\},
\end{align*}
\]  \tag{C-1}

where with FC-R-MPC, \(J^s(k_c)\) is defined by the global objective function in Equation (3-1) and, with DC-R-MPC, \(J^s(k_c)\) is defined by the local objective function in Equation (3-2). Furthermore, with FC-R-MPC, \(\hat{U}_r^s\) and \(\hat{U}_{VSL}^s\) contain all the intermediate solutions of the other agents in the network and with DC-R-MPC, \(\hat{U}_r^s\) and \(\hat{U}_{VSL}^s\) contain the intermediate solutions of agent \(s + 1\). Moreover, \(x\) and \(d\) contain the states and the demands of the comprehensive network with FC-R-MPC, and the states and the demands of subsystems \(s\) and \(s + 1\) with DC-R-MPC.

### C-2-2 Continuous optimisation algorithm details

SQP in combination with a multi-start approach is used to solve the nonlinear non-convex continuous optimisation problems, similar to the method that is used with the proposed algorithms, described in Appendix C-1-2. However, the solution space is larger, as one RM installation and two VSLs are considered in the optimisation problems instead of only one RM installation. Hence, more initial point profiles need to be evaluated with the multi-start approach. All possible permutations of the values 0, 0.5, 1 for the RM rates and 40, 75, 100 for the VSLs, constant over the horizon, are used as initial point profiles. Hence 9 initial point profiles are evaluated with the multi-start approach.

The \texttt{fmincon} function of the Optimization Toolbox of MATLAB is used to solve the continuous optimisation problems with SQP. The default SQP optimisation options are used, with some modifications:

- The setting \texttt{MaxFunctionEvaluations} is set to \(10 \cdot 10^6\),
- The setting \texttt{MaxIterations} is set to \(10 \cdot 10^6\).
C-3 Cent-A-MPC

Cent-A-MPC uses an alternating optimisation scheme to optimise the global objective function defined in Equation (3-1), similar to FC-A-MPC. However, where with FC-A-MPC multiple agents optimise this objective function for local control inputs in a distributed manner, with Cent-A-MPC one single agent optimises the global objective function for all the control inputs in the network. After \( n_{alt} \) iterations, the global objective function values are evaluated with the found inputs in every iteration. The first time samples of the inputs that yield the lowest global objective function value are sent to the network.

In the remainder of this section, both the continuous optimisation problems and the discrete optimisation problems for Cent-A-MPC are formulated and the methods that are used to solve the problems are discussed.

C-3-1 Problem formulation

The nonlinear non-convex continuous optimisation problem for the centralised algorithm is formulated as:

\[
\min_{\bar{U}_r(k_c)} \tilde{J}_{\text{con}}(k_c) \tag{C-3}
\]

subject to:

\[
x(k_c + 1) = f_M((x(k_c + l), \bar{U}_r(k_c + l), \bar{U}_{\text{VSL}}(k_c + l), d(k_c + l)),
\]

\[
x(k_c) = x_k,
\]

\[
d(k_c + l) = d_k,
\]

\[
U^i_r(k_c + 1) \in \mathcal{U}_r, \forall i \in I_r,
\]

\[
U^i_r(k_c + j) = U^i_r(k_c + N_c - 1), \forall i \in I_r,
\]

for \( l \in \{0, 1, \ldots, N_p - 1\} \),

\[
\text{for } j \in \{N_u, \ldots, N_p - 1\},
\]

where \( \tilde{J}_{\text{con}}(k_c) \) is defined as:

\[
\tilde{J}_{\text{con}}(k_c) = M(k_c + N_p) \left[ T_c \left( \sum_{i \in I_{\text{all}}} \rho_i(k) L_i \lambda + \sum_{i \in I_r} w_i(k) \right) + \right. \\
\left. \zeta_w \left( \sum_{i \in I_r} \left( \max(w_i(k) - w_{\text{max}}, 0) \right)^2 \right) \right] + \\
\zeta_r \sum_{j=1}^{N_p-1} \left( \sum_{i \in I_r} \left( U^i_r(k_c + j) - U^i_r(k_c + j - 1) \right)^2 \right), \tag{C-5}
\]

\( \bar{U}_r \) contains the optimal RM rates for all the RM installations in the network, \( \bar{U}_{\text{VSL}} \) contains all the intermediately found VSLs in the discrete optimisation, and \( x \) and \( d \) contain the states and the demands of the comprehensive network.
The discrete optimisation problem for the centralised algorithm is formulated as:

$$\min_{\bar{U}_{\text{VSL}}(k_c)} \bar{J}_{\text{dis}}(k_c)$$

subject to:

$$x(k_c + l + 1) = f_M((x(k_c + l), \bar{U}_{\text{VSL}}(k_c + l), \hat{U}_r(k_c + l), d(k_c + l)), $$

$$x(k_c) = x_k,$$

$$d(k_c + l) = d_k,$$

$$U^i_{\text{VSL}}(k_c + l) \in \mathcal{U}_{\text{VSL}}, \forall i \in I_{\text{VSL}},$$

$$U^i_{\text{VSL}}(k_c + j) = U^i_{\text{VSL}}(k_c + N_c - 1), \forall i \in I_{\text{VSL}},$$

for $$l \in \{0, 1, \ldots, N_p - 1\},$$

for $$j \in \{N_u, \ldots, N_p - 1\},$$

where $$\bar{J}_{\text{dis}}(k_c)$$ is defined in Equation (3-10), $$\bar{U}_{\text{VSL}}$$ contains the optimal VSLs for all the matrix signs in the network and $$\hat{U}_r$$ contains all the intermediately found RM rates in the continuous optimisation.

### C-3-2 Discrete optimisation algorithm details

The number of feasible VSL combination grows exponentially with the size of the problem. Hence, it is not possible to evaluate all the feasible solutions to solve the discrete optimisation problems of Cent-A-MPC in a reasonable time. Therefore, a GA is used to solve the discrete optimisation problems of the centralised algorithm.

The function $$ga$$ of the Global Optimization Toolbox of MATLAB is used to solve the discrete optimisation problems with a GA. The default options of the toolbox are used with some modifications:

- The setting $$\text{PopulationSize}$$ is set to 800,
- The setting $$\text{MaxStallGenerations}$$ is set to 400.

### C-3-3 Continuous optimisation algorithm details

SQP in combination with a multi-start approach is used to solve the nonlinear non-convex continuous optimisation problems of the centralised algorithm, similar to the method that is used with the proposed algorithms, described in Appendix C-1-2. However, the solution space is significantly larger, as three RM installations are considered as control inputs in the optimisation. Therefore, more initial point profiles need to be evaluated with the multi-start approach. All possible permutations of the values 0.1, 0.45 and 0.8, constant over the horizon, are used as initial point profiles. Additionally, 10 random initial point profiles are used to replace the linearly decreasing and linearly increasing profiles described in Appendix C-1-2. These are replaced because including those in the permutation would lead to a very large number of initial point profiles. Hence, a total of 37 initial point profiles are evaluated with the multi-start approach.
The `fmincon` function of the Optimization Toolbox of MATLAB is used to solve the continuous optimisation problems with SQP. The default SQP optimisation options are used, with some modifications:

- The setting `MaxFunctionEvaluations` is set to $10 \cdot 10^6$,
- The setting `MaxIterations` is set to $10 \cdot 10^6$.

## C-4 Dec-A-MPC

Similar to the proposed algorithms, the network is partitioned into subsystems controlled by local agents with Dec-A-MPC. However, the agents do not cooperate with the decentralised algorithm. Hence, all agents optimise a local objective function that only contains the states and inputs of the local subsystem. The decentralised algorithm uses an alternating optimisation scheme to relax the initial MINLP optimisation problems into continuous and discrete optimisation problems.

The remainder of this section formalises the discrete optimisation problems and continuous optimisation problems for Dec-A-MPC. Moreover, the methods that are used to solve the problems are described.

### C-4-1 Problem formulation

The continuous optimisation problem of agent $s$ is formulated as:

$$\min_{\hat{U}^s_{\text{con}}(k_c)} J^s_{\text{con}}(k_c)$$  \hspace{1cm} (C-8)

subject to:

$$x(k_c + l + 1) = f_M((x(k_c + l), \hat{U}^s_{\text{con}}(k_c + l), \hat{U}^s_{\text{VSL}}(k_c + l), d(k_c + l)), \hspace{1cm} (C-9)$$

$$x(k_c) = x_k,$$

$$d(k_c + l) = d_k,$$

$$U^i_l(k_c + l) \in \mathcal{U}, \forall i \in I^s_i,$$

$$U^i_l(k_c + j) = U^i_l(k_c + N_c - 1), \forall i \in I^s_i,$$

for $l \in \{0, 1, \ldots, N_p - 1\},$

for $j \in \{N_u, \ldots, N_p - 1\},$

where $J^s_{\text{con}}(k_c)$ is defined as:

$$J^s_{\text{con}}(k_c) = \sum_{k=Mk_c}^{M(k_c+N_p)} \left[ T_c \left( \sum_{i \in I^s_i} \rho_i(k)L_i \lambda + \sum_{i \in I^s_i} w_i(k) \right) + \zeta_w \left( \sum_{i \in I^s_i} \left( \max(w_i(k) - w_{\text{max}}, 0) \right)^2 \right) \right] + \hspace{1cm} (C-10)$$

$$+ \zeta_r \left[ \sum_{j=1}^{N_p-1} \left( \sum_{i \in I^s_i} \left( U^i_l(k_c + j) - U^i_l(k_c + j - 1) \right)^2 \right) \right]$$
\( \hat{U}_{VSL}^s \) contains all the intermediately found VSLs in the discrete optimisation, and \( x \) and \( d \) contain the states and demands of subsystem \( s \).

The discrete optimisation problem of agent \( s \) is formulated as:

\[
\begin{align*}
\min_{\tilde{U}_{VSL}^s(k_c)} & \quad J_{\text{dis}}^s(k_c) \\
\text{subject to:} & \\
& \quad x(k_c + l + 1) = f_M((x(k_c + l), \hat{U}_{VSL}^s(k_c + l), \hat{U}_r^s(k_c + l), d(k_c + l)), \\
& \quad x(k_c) = x_k, \\
& \quad d(k_c + l) = d_k, \\
& \quad U_{VSL}^i(k_c + l) \in \mathcal{U}_{VSL}, \forall i \in I_{VSL}^s, \\
& \quad U_{VSL}^i(k_c + j) = U_{VSL}^i(k_c + N_c - 1), \forall i \in I_{VSL}^s, \\
& \quad \text{for } l \in \{0, 1, \ldots, N_p - 1\}, \\
& \quad \text{for } j \in \{N_u, \ldots, N_p - 1\},
\end{align*}
\]

where \( J_{\text{dis}}^s(k_c) \) is defined as:

\[
J_{\text{dis}}^s(k_c) = \sum_{k=Mk_c}^{M\{k_c+N_p\}} \left[ T_c \left( \sum_{i \in I_s} p_i(k) L_i \lambda + \sum_{i \in I_r^s} w_i(k) \right) + \right. \\
\left. \zeta_w \left( \sum_{i \in I_r^s} \left( \max(w_i(k) - w_{\text{max}}, 0) \right)^2 \right) \right],
\]

and \( \hat{U}_r^s \) contains all the intermediately found VSLs in the discrete optimisation of the decentralised algorithm.

**C-4-2 Discrete optimisation algorithm details**

The discrete optimisation problems are solved by evaluating all possible solutions of the search tree. The solution space of the feasible VSL combinations is the same as with the proposed algorithms. Hence, the same method is used to solve the optimisation problems, described in Appendix C-1-1.

**C-4-3 Continuous optimisation algorithm details**

The nonlinear non-convex continuous optimisation problems are solved by combining SQP with a multi-start approach. The solution space of the continuous optimisation problems is the same as with the proposed algorithms. Hence, the same method is used to solve the optimisation problems, described in Appendix C-1-2.
This appendix provides supportive figures for the case study described in Section 4.

D-1  Case study A: Relaxation on VSLs

This section contains supportive figures that describe the states and control inputs of the simulations in case study A that are described in Section 4-3.

D-1-1  FC-R-MPC with \( n_{\text{dist}} = 1 \)

The simulation results of FC-R-MPC with \( n_{\text{dist}} = 1 \) are shown in this section. The relaxation on the discrete nature of the VSLs results in a significant system performance loss. Hence, a significant traffic jam originates in front of the third on-ramp, as can be seen in Figure D-1. This results in a system performance of \( TTS = 4679 \text{ veh-h} \).

![Figure D-1: A heat map of the densities of all the segments when FC-R-MPC controls the network with \( n_{\text{dist}} = 1 \).](image-url)
Figure D-2: A heat map of the average speeds of all the segments when FC-R-MPC controls the network with $n_{\text{dist}} = 1$.

Figure D-3: The queue lengths in front of the three on-ramps when FC-R-MPC controls the network with $n_{\text{dist}} = 1$.

Figure D-4: The metering rates of the three RM installations when FC-R-MPC controls the network with $n_{\text{dist}} = 1$. 
D-1 Case study A: Relaxation on VSLs

D-1-2 FC-R-MPC with $n_{\text{dist}} = 4$

The simulation results of FC-R-MPC with $n_{\text{dist}} = 4$ are shown in this section. The relaxation on the discrete nature of the VSLs results in a significant system performance loss. Hence, a traffic jam originates in front of the third on-ramp, as can be seen in Figure D-6. This results in a system performance of $TTS = 4587$ veh·h.

Figure D-5: The values of the six VSLs when FC-R-MPC controls the network with $n_{\text{dist}} = 1$.

Figure D-6: A heat map of the densities of all the segments when FC-R-MPC controls the network with $n_{\text{dist}} = 4$.

Figure D-7: A heat map of the average speeds of all the segments when FC-R-MPC controls the network with $n_{\text{dist}} = 4$. 
Supportive Figures

**Figure D-8:** The queue lengths in front of the three on-ramps when FC-R-MPC controls the network with $n_{\text{dist}} = 4$.

**Figure D-9:** The metering rates of the three RM installations when FC-R-MPC controls the network with $n_{\text{dist}} = 4$.

**Figure D-10:** The values of the six VSLs when FC-R-MPC controls the network with $n_{\text{dist}} = 4$.

D-1-3 DC-R-MPC with $n_{\text{dist}} = 1$

The simulation results of DC-R-MPC with $n_{\text{dist}} = 1$ are shown in this section. The relaxation on the discrete nature of the VSLs results in a significant system performance loss. Hence, a
significant traffic jam originates in front of the third on-ramp, as can be seen in Figure D-11. This results in a system performance of \(TTS = 4685\) veh-h.

**Figure D-11**: A heat map of the densities of all the segments when DC-R-MPC controls the network with \(n_{\text{dist}} = 1\).

**Figure D-12**: A heat map of the average speeds of all the segments when DC-R-MPC controls the network with \(n_{\text{dist}} = 1\).

**Figure D-13**: The queue lengths in front of the three on-ramps when DC-R-MPC controls the network with \(n_{\text{dist}} = 1\).
**Figure D-14:** The metering rates of the three RM installations when DC-R-MPC controls the network with $n_{\text{dist}} = 1$.

**Figure D-15:** The values of the six VSLs when DC-R-MPC controls the network with $n_{\text{dist}} = 1$.

**D-1.4 DC-R-MPC with $n_{\text{dist}} = 4$**

The simulation results of DC-R-MPC with $n_{\text{dist}} = 4$ are shown in this section. The relaxation on the discrete nature of the VSLs results in a significant system performance loss. Hence, a significant traffic jam originates in front of the third on-ramp, as can be seen in Figure D-16. This results in a system performance of $TTS = 4612$ veh·h.
**Figure D-16:** A heat map of the densities of all the segments when DC-R-MPC controls the network with $n_{\text{dist}} = 4$.

**Figure D-17:** A heat map of the average speeds of all the segments when DC-R-MPC controls the network with $n_{\text{dist}} = 4$.

**Figure D-18:** The queue lengths in front of the three on-ramps when DC-R-MPC controls the network with $n_{\text{dist}} = 4$. 
Figure D-19: The metering rates of the three RM installations when DC-R-MPC controls the network with $n_{\text{dist}} = 4$.

Figure D-20: The values of the six VSLs when DC-R-MPC controls network with $n_{\text{dist}} = 4$.

D-1-5 FC-A-MPC with $n_{\text{dist}} = 1$

The simulation results of FC-A-MPC with $n_{\text{dist}} = 1$ are shown in this section. The algorithm has a good system performance by considering the VSLs as discrete decision variables in the optimisation. Hence, the network remains mostly uncongested, as can be seen in Figure D-21. However, the soft constraint at the third on-ramp is violated significantly, partly due to the small number of distributed iterations. The queue lengths are illustrated in Figure D-23. The algorithm has a system performance of $\text{TTS} = 4481 \text{ veh-h}$. 
Figure D-21: A heat map of the densities of all the segments when FC-A-MPC controls the network with $n_{\text{dist}} = 1$.

Figure D-22: A heat map of the average speeds of all the segments when FC-A-MPC controls the network with $n_{\text{dist}} = 1$.

Figure D-23: The queue lengths in front of the three on-ramps when FC-A-MPC controls the network with $n_{\text{dist}} = 1$. 
Figure D-24: The metering rates of the three RM installations when FC-A-MPC controls the network with $n_{\text{dist}} = 1$.

Figure D-25: The values of the six VSLs when FC-A-MPC controls the network with $n_{\text{dist}} = 1$.

D-1-6  FC-A-MPC with $n_{\text{dist}} = 4$

The simulation results of FC-A-MPC with $n_{\text{dist}} = 4$ are shown in this section. The algorithm has a good system performance by considering the VSLs as discrete decision variables in the optimisation. Hence, the network remains mostly uncongested, as can be seen in Figure D-26. The soft constraint at the third on-ramp are slightly violated but remains within 10% of the limit. The queue lengths are illustrated in Figure D-28. The algorithm has a system performance of $TTS = 4466$ veh·h.
Figure D-26: A heat map of the densities of all the segments when FC-A-MPC controls the network with $n_{dist} = 4$.

Figure D-27: A heat map of the average speeds of all the segments when FC-A-MPC controls the network with $n_{dist} = 4$.

Figure D-28: The queue lengths in front of the three on-ramps when FC-A-MPC controls the network with $n_{dist} = 4$. 
**Figure D-29:** The metering rates of the three RM installations when FC-A-MPC controls the network with $n_{\text{dist}} = 4$.

**Figure D-30:** The values of the six VSLs when FC-A-MPC controls the network with $n_{\text{dist}} = 4$.

### D-1-7 DC-A-MPC with $n_{\text{dist}} = 1$

The simulation results of DC-A-MPC with $n_{\text{dist}} = 1$ are shown in this section. As the algorithm uses a small number of distributed iterations, a significant traffic jam originates in front of the third on-ramp as can be seen in Figure D-31. The queue lengths are slightly violated but remain within 10% of the soft constraint limit. The queue lengths are illustrated in Figure D-33. The algorithm has a system performance of $\text{TTS} = 4515 \text{ veh-h}$. 
**Figure D-31:** A heat map of the densities of all the segments when DC-A-MPC controls the network with $n_{\text{dist}} = 1$.

**Figure D-32:** A heat map of the average speeds of all the segments when DC-A-MPC controls the network with $n_{\text{dist}} = 1$.

**Figure D-33:** The queue lengths in front of the three on-ramps when DC-A-MPC controls the network with $n_{\text{dist}} = 1$. 

Master of Science Thesis  
Ugiješa Todorović
Figure D-34: The metering rates of the three RM installations when DC-A-MPC controls the network with \( n_{\text{dist}} = 1 \).

Figure D-35: The values of the six VSLs when DC-A-MPC controls the network with \( n_{\text{dist}} = 1 \).

D-1-8 DC-A-MPC with \( n_{\text{dist}} = 4 \)

The simulation results of DC-A-MPC with \( n_{\text{dist}} = 4 \) are shown in this section. A traffic jam originates in front of the third on-ramp as can be seen in Figure D-36. The algorithm has a system performance of \( \text{TTS} = 4505 \text{ veh} \cdot \text{h} \).

Figure D-36: A heat map of the densities of all the segments when DC-A-MPC controls the network with \( n_{\text{dist}} = 4 \).
Figure D-37: A heat map of the average speeds of all the segments when DC-A-MPC controls the network with $n_{\text{dist}} = 4$.

Figure D-38: The queue lengths in front of the three on-ramps when DC-A-MPC controls the network with $n_{\text{dist}} = 4$.

Figure D-39: The metering rates of the three RM installations when DC-A-MPC controls the network with $n_{\text{dist}} = 4$. 
Figure D-40: The values of the six VSLs when DC-A-MPC controls the network with $n_{\text{dist}} = 4$.

D-2 Case study B: Architecture comparison

This section contains supportive figures for the simulations in case study B, described in Section 4-4.

D-2-1 Cent-A-MPC

A heat map of the average speeds in the network with Cent-A-MPC is shown in Figure D-41. The computational times per controller sample are shown in Figure D-42.

Figure D-41: A heat map of the average speed of all the segments when Cent-A-MPC controls the network.
D-2 Case study B: Architecture comparison

Figure D-42: The computational time for every controller sample when Cent-A-MPC controls the network.

D-2-2 Dec-A-MPC

A heat map of the average speeds in the network with Dec-A-MPC is shown in Figure D-43.

Figure D-43: A heat map of the average speed of all the segments when Dec-A-MPC controls the network.
**D-2-3 FC-A-MPC**

A heat map of the average speeds in the network with FC-A-MPC is shown in Figure D-44.

**Figure D-44:** A heat map of the average speed of all the segments when FC-A-MPC controls the network.

**D-2-4 DC-A-MPC**

A heat map of the average speeds in the network with DC-A-MPC is shown in Figure D-45.

**Figure D-45:** A heat map of the average speed of all the segments when DC-A-MPC controls the network.
This appendix presents a paper that is written on the main contributions of this thesis to the state-of-the-art. The paper is written in the format of IEEE Transactions on Intelligent Transportation Systems.
Distributed MPC for Large Freeway Networks using Alternating Optimization

Uglješa Todorović, José Ramón D. Frejo, Member, IEEE, and Bart De Schutter, Senior Member, IEEE

Abstract—The Model Predictive Control (MPC) framework has shown outstanding capabilities for the control of Variable Speed Limits (VSLs) and Ramp Metering (RM) installations. However, implementation to large freeway networks is challenging, as a Mixed Integer Nonlinear Programming (MINLP) optimization problem has to be solved within every controller sampling interval. Consequently, many related papers relax the MINLP problems by considering the VSLs to be continuous decision variables. This paper proposes two novel MPC algorithms for coordinated control of discrete VSLs and continuous RM rates that do not make this relaxation. The proposed algorithms use a distributed control architecture and an alternating optimization scheme to relax the MINLP optimization problems and, hence, offer a trade-off between computational complexity and system performance. The performance of the proposed algorithms is evaluated in a case study. The case study shows that relaxing the VSLs to be continuous decision variables with a distributed architecture results in a significant performance loss. Moreover, the proposed algorithms are compared to the more conventional centralized and decentralized algorithms. Both proposed algorithms have a lower computational complexity than the centralized algorithm, as they manage to solve all optimization problems within the sampling intervals. Moreover, one of the proposed algorithms has a system performance that is similar to the optimal performance of the centralized algorithm.

Index Terms—Alternating optimization, distributed MPC, freeway traffic control, ramp metering, variable speed limits.

I. INTRODUCTION

As the number of vehicles and the need for transportation keeps increasing every year, traffic congestion has become a crucial problem in today’s society. There is a need for a sustainable solution to reduce or even eliminate traffic jams. Freeway traffic control has shown to be a sustainable solution to this problem [1], [2], [3].

Especially the implementation of Ramp Metering (RM) installations and Variable Speed Limits (VSLs) as control measures is currently a widely researched area, because the proper coordination of those measures can significantly reduce traffic congestion, traffic emissions and the risk of accidents [4], [5]. However, these control measures have to be coordinated at a large spatial scale for good performance, as local control inputs have significant influences on the traffic state of distant parts of the freeway [6].

The Model Predictive Control (MPC) [7] framework has shown outstanding capabilities for the proper coordination of those control measures [2], [4], [5], but implementation to large freeway networks remains challenging. Mostly nonlinear prediction models are used in literature (e.g. in: [3], [4], [8]) due to the inherent nonlinearity of traffic flow. Moreover, by law, VSLs are only allowed to take prescribed discrete values, while RM rates are continuous inputs. The combination of a nonlinear prediction model and discrete VSLs with continuous RM rates yields a Mixed Integer Nonlinear Programming (MINLP) optimization problem that has to be solved within every controller sampling interval. The computational time that is needed to solve such an optimization problem generally increases exponentially with the size of the problem (i.e. increases with the number of control measures, the control horizon, etc.). Hence, with the application of MPC to large freeway networks, the computational time quickly becomes larger than the controller sampling time, making it unimplementable.

Various approaches have been considered to make MPC implementable to large freeway networks. Some approaches simplify the nonlinear prediction model by rewriting it as an MLD model [9], relaxing the MINLP optimization problems into Mixed Integer Linear Programming (MILP) optimization problems [1], [10]. Other approaches parametrize the control signals into control laws to simplify the optimization problems [11]. Some approaches consider the VSLs as continuous decision variables in the optimization [4], [5], relaxing the MINLP optimization problems into nonlinear continuous optimization problems. However, related works have contradictory conclusions on this relaxation, as some have found that it results in a large performance loss [12]. Hence, the effect of this relaxation on the system performance is still an open question in the field and is investigated in this paper.

A distributed control architecture [13] is a popular approach to reduce the computational complexity of large-scale MPC problems in general. However, the implementation of such an architecture to freeway networks is difficult, as every distributed agent still has to solve an MINLP problem. Hence, the existing distributed approaches with application to freeway traffic control either do not consider VSLs as control inputs [1], [14], [15], or consider these to be continuous decision variables in the optimization [5], [6].

The main contribution of this paper is the proposal of two novel MPC algorithms that use a distributed control architecture and an alternating optimization scheme for coordinated control of discrete VSLs and continuous RM rates: Fully Cooperative Alternating Model Predictive Control (FC-A-MPC) and Downstream Cooperative Alternating Model Predictive Control (DC-A-MPC). Both proposed algorithms are similar, but offer a different trade-off between computational complexity and system performance. Moreover, both algorithms reduce the computational complexity of the problem sufficiently, such that they are implementable in real-time for large freeway
networks. Furthermore, this paper contributes to the state-of-the-art by investigating the effects on the system performance by relaxing the VSLs to be continuous decision variables in a distributed setting.

The remainder of this paper is structured as follows. Section II outlines the components of the MPC framework that are used in this work. Subsequently, Section III proposes the two novel MPC algorithms. Then, Section IV presents a case study that evaluates the proposed algorithms. Finally, Section V concludes this paper.

II. MPC FOR FREEWAY TRAFFIC

A. Introduction

MPC is an advanced control methodology where an objective function is optimized to find optimal control inputs and a prediction model is used to predict relevant future system trajectories. With the optimization, multiple control inputs can be determined, and the prediction model can be used to predict influences of those control inputs on distant parts of the network. Therefore, the MPC framework is highly suitable for the coordination of VSLs and RM installations.

The various MPC approaches mainly differ in the prediction models, the objective functions and the control architectures. Therefore, the rest of this section outlines the prediction model, objective function and control architectures that are used in this work. For a detailed description of MPC, the reader is referred to [7].

B. Prediction model

The macroscopic traffic model METANET [16] is used as the prediction model, as it provides a good trade-off between computational complexity and model accuracy [17] and is capable of modeling RM installations and VSLs as control inputs. It is a second-order nonlinear model, which models traffic flow analogous with a compressible fluid. Consequently, the computational speed of METANET is not affected by vehicular density, making it highly suitable for model-based control.

A traffic network is modeled as a directed graph, where the links correspond to homogeneous freeway stretches with similar geometry. Each link $m$ is partitioned into segments $i$ of length $L_m,i$. The traffic state of each segment $i$ of link $m$ at time $k$ is described by three aggregated variables: the traffic density $p_{m,i}(k)$ (in veh/km), the space-mean speed $v_{m,i}(k)$ (in km/h) and the traffic flow $q_{m,i}(k)$ (in veh/h). The rest of this work only differentiates between segments to improve readability. This is possible because the network in the case study can be modeled by one large link.

The outflow $q_i(k)$ of segment $i$ during period $[kT, (k+1)T]$ is an auxiliary variable and is described by:

$$ q_i(k) = \lambda \rho_i(k) v_i(k), \quad (1) $$

where $\lambda$ is the number of lanes in the network and $T$ is the sampling time related to the time $t$ and the sample number $k$ by $t = kT$. The first system equation expresses the conservation of vehicles in a segment:

$$ \rho_i(k+1) = \rho_i(k) + \frac{\tau}{\lambda L_i} \left[ (1 - \beta_i(k)) q_{i-1}(k) - q_i(k) + q_{r,i}(k) \right], \quad (2) $$

where $\beta_i(k)$ is the fraction of the flow that leaves segment $i$ from an off-ramp and $q_{r,i}(k)$ is the flow that enters segment $i$ from an on-ramp. Hence, if segment $i$ does not contain an on-ramp or off-ramp, $\beta_i(k)$ or $q_{r,i}(k)$ are, respectively, zero.

The second system equation describes the speed dynamics in a heuristic way. Consequently, the update equation for the space-mean speed $v_i(k)$ involves a relaxation term that expresses the drivers will to achieve a desired speed $V(\rho_i(k))$, a convection term that describes the change in speed that is caused by the inflow of upstream vehicles and an anticipation term that expresses the change in speed caused by a difference in density with respect to the downstream segment:

$$ v_i(k+1) = v_i(k) + \frac{T}{L_i} V(\rho_i(k)) - v_i(k) - \frac{\eta_T}{\tau L_i} \left[ \rho_{i+1}(k) - \rho_i(k) \right], \quad (3) $$

where $\tau$, $\kappa$ and $\eta$ are model parameters and the desired speed is described by:

$$ V(\rho_i(k)) = v_{free} \exp \left[ -\frac{1}{a} \left( \frac{\rho_i(k)}{\rho_{crit}} \right)^a \right]. \quad (4) $$

where $a$ is a model parameter, $v_{free}$ is the free-flow speed and $\rho_{crit}$ is the critical density. If segment $i$ contains an on-ramp, the following term is added to (3) to model the merging phenomena:

$$ \frac{\delta T q_{r,i}(k) v_i(k)}{L_i \lambda (\rho_i(k) + \kappa)}, \quad (5) $$

where $\delta$ is a model parameter.

Generally, drivers tend to decelerate faster than they tend to accelerate. Hence, it is suggested in [4] that the anticipation term $\eta$ in (3) takes different values for when the downstream density is higher or lower than the density in segment $i$:

$$ \eta = \begin{cases} \eta_h & \text{if } \rho_{i+1}(k) \geq \rho_i(k) \\ \eta_l & \text{otherwise,} \end{cases} \quad (6) $$

where $\eta_h$ and $\eta_l$ are anticipation terms for the two different cases.

The vehicles that are waiting at an on-ramp are modeled with a vertical queue model, where the dynamics of the queue length $w_i(k)$ at the on-ramp of segment $i$ are described by:

$$ w_i(k+1) = w_i(k) + T [d_i(k) - q_{r,i}(k)], \quad (7) $$
where \( d_i(k) \) is the on-ramp demand and the on-ramp flow \( q_{r,i}(k) \) is described by:

\[
q_{r,i}(k) = \min \left[ d_i(k) + w_i(k), C U_i^r(k), C \left( \frac{\rho_{\text{max}}}{\rho_{\text{crit}}} - \frac{\rho_i(k)}{\rho_{\text{crit}}} \right) \right],
\]

where \( C \) is the on-ramp capacity, \( \rho_{\text{max}} \) is the jam density of the freeway and \( U_i^r(k) \), \( i \in [0, 1] \) is the metering rate, which is a control input. Hence, the metering rate is the percentage of vehicles that may enter the freeway.

The VSLs are modeled according to [4] by modifying (4) for the segments that contain an on-ramp to:

\[
V(\rho_i(k)) = \min \left[ v_{\text{free}} \exp \left( -\frac{1}{\alpha} \left( \frac{\rho_i(k)}{\rho_{\text{crit}}} \right)^\alpha \right), \right. \]

\[
\left. (1 + \alpha) U_{VSL}^i(k) \right],
\]

where \( \alpha \) is a non-compliance factor of drivers and \( U_{VSL}^i(k) \) is the VSL at segment \( i \), which is a control input. Other approaches for modeling VSLs are presented in [18].

For a more detailed description of METANET, the reader is referred to [16].

C. Objective function

This paper uses three sub-objectives, which are added together with appropriate weights for the comprehensive network objective function.

1) Congestion reduction: The Total Time Spent (TTS) is used as the performance criterion to minimize the congestion in the network. This is the total time that vehicles spend in a section and is the most commonly used criterion for the reduction of congestion (e.g. in: [1], [5], [14], [17]).

2) Soft constraints: The queue lengths at the on-ramps have to be limited to avoid spillback to urban roads. This work uses a term in the objective functions that imposes soft constraints on these queue lengths. This approach has been suggested in [17], as sometimes a slight violation in queue lengths can result in an overall much better global system performance.

3) Signal fluctuation: A term is included to penalize the variation in metering rates to avoid large and frequent fluctuations in the control signal. This is common practice in the field (e.g. in: [4], [5], [15], [19]).

Subsequently, the global objective function \( J(k_c) \), describing the network performance at time step \( k_c \) over the prediction horizon \( N_p \), is given by:

\[
J(k_c) = \sum_{k = MK_c}^{M(k_c + N_p)} \left[ T_c \left( \sum_{i \in I_4} \rho_i(k)L_i + \sum_{i \in I_1} w_i(k) \right) + \right. \]

\[
\zeta_w \left( \sum_{i \in I_4} \left( \max(w_i(k) - w_{\text{max}}(0)) \right)^2 \right) + \]

\[
\sum_{j=1}^{N_i - 1} \left[ \zeta_{\text{q}} \left( \sum_{i \in I_4} \left( U_i^e(k_c + j) - U_i^e(k_c + j - 1) \right)^2 \right) \right],
\]

where \( M \) relates the control time step \( k_c \) and simulation time step \( k \) as \( k = MK_c \). \( T_c \) is the sampling time of the controller, \( \zeta_w \) and \( \zeta_{\text{q}} \) are weighting terms, \( w_{\text{max}} \) is the soft limit on the queue lengths, \( I_4 \) is the set of all segments of the network and \( I_1 \) is the set of segments in the network that contain a controlled on-ramp.

D. Control architectures

The two proposed distributed algorithms are evaluated by comparing their performance to the more conventional centralized and decentralized algorithms.

1) Centralized MPC: With a centralized architecture, one central MPC agent determines control inputs for the comprehensive network. The inputs are found by optimizing the global objective function using measurements of all the states and, consequently, all control measures are coordinated optimally. Hence, centralized MPC generally leads to the optimal system performance in a receding horizon context [1], [5], [14], [15]. However, it also has the largest computational complexity, as the size of the optimization problems scale with the size of the considered network.

2) Decentralized MPC: As opposed to a centralized architecture, with a decentralized architecture, the network is partitioned into subsystems and each subsystem is controlled by a local MPC agent. The agents determine local control inputs by optimizing a local objective function without communicating with each other. Consequently, decentralized MPC has the lowest computational complexity, which only scales with the size of the subsystems. However, generally, it also has the worst system performance in a receding horizon context [1], [5], [14], [15], as the control inputs are not coordinated at all.

3) Distributed MPC: An intermediate solution to the drawbacks of a centralized and decentralized architecture is a distributed architecture. Similar to a decentralized architecture, the network is partitioned into subsystems and each subsystem is controlled by a local MPC agent. However, the agents are now actively communicating with each other to improve the global network performance.

Since distributed MPC is a generic term for all approaches where agents at least share some information while determining control inputs, there is a great variety in distributed algorithms [20], [21]. However, with freeway traffic control, agents often share their objective function with other agents, such as the downstream neighboring agent [1], [15] or every other agent in the network [5], [15], but optimize this objective function for local control inputs. Once the agents solve their optimization problems, they communicate their intermediate solutions to the other agents, and proceed solving their optimization problems with the updated solutions of the other agents.

III. DISTRIBUTED MPC USING ALTERNATING OPTIMIZATION

This section proposes two novel MPC algorithms, FC-A-MPC and DC-A-MPC, for coordinated control of discrete...
VSLs and continuous RM rates. Both algorithms use a distributed control architecture and an alternating optimization scheme to relax the MINLP problems. The remainder of this section formalizes both algorithms.

A. Distributed architecture

An arbitrary long freeway network is considered that is modeled by $N_{\text{all}}$ segments $i \in I_{\text{all}}$. All segments $i \in I_{\text{off}}$ contain an on-ramp and all segments $i \in I_{\text{on}}$ contain an off-ramp. All on-ramps have an RM installation that can control the number of vehicles that enter the network. Furthermore, the segments $i \in I_{\text{VSL}}$ contain a matrix sign that can display speed limits. The network is subject to a mainstream demand $d_0$ and on-ramp demands $d_i$, $\forall i \in I_i$.

For the distributed architecture, the freeway is partitioned into $N_{\text{sub}}$ subsystems, where each subsystem contains an arbitrary number of VSLs and RM installations. Subsequently, $I_{\text{all}}$ is partitioned for every subsystem $s$ in $I_s$, such that all segments in $I_s$ are part of subsystem $s$. Similarly, $I_s$, $I_{\text{VSL}}$, $I_{\text{off}}$ are partitioned in $I_{s}^l$, $I_{s,\text{VSL}}$, $I_{s,\text{off}}$ for every subsystem $s$. Associated with the subsystems are $N_{\text{sub}}$ MPC agents, that determine local control inputs for their respective subsystem.

The two proposed algorithms differ in cooperativeness. FC-A-MPC has a fully cooperative architecture [5], where all agents share the same global objective function, but optimize it for local decision variables. Hence, the global objective function, based on (10) but partitioned for agent $s$, is formulated as:

$$
\begin{align*}
J^s(k_c) &= \sum_{k=Mk_c}^{M(k_c+N_p)} \left[ T_c \left( \sum_{i \in I_{s}} \rho_i(k_c) L_i \lambda_s + \sum_{i \in I_{s}} w_i(k) \right) \right. \\
&\quad \left. + \zeta_w \left( \sum_{i \in I_{s}} \left( \max(w_i(k) - w_{\text{max}}, 0) \right)^2 \right) \right] + \\
&\quad \sum_{j=1}^{N_p-1} \left[ \zeta_u \left( \sum_{i \in I_{s}} \left( U_i(k_c + j) - U_i(k_c + j - 1) \right)^2 \right) \right].
\end{align*}
$$

(11)

DC-A-MPC has a downstream cooperative architecture [1], [15], where every agent $s$ optimizes a local objective function that contains the states of agent $s$ and $s+1$. The local objective function of agent $s$, based on (10), is formulated as:

$$
\begin{align*}
J^s(k_c) &= \sum_{k=Mk_c}^{M(k_c+N_p)} \left[ T_c \left( \sum_{i \in I_{s}} \rho_i(k_c) L_i \lambda_s + \sum_{i \in I_{s}} w_i(k) \right) \right. \\
&\quad \left. + \zeta_w \left( \sum_{i \in I_{s}} \left( \max(w_i(k) - w_{\text{max}}, 0) \right)^2 \right) \right] + \\
&\quad \sum_{j=1}^{N_p-1} \left[ \zeta_u \left( \sum_{i \in I_{s}} \left( U_i(k_c + j) - U_i(k_c + j - 1) \right)^2 \right) \right].
\end{align*}
$$

(12)

A parallel and iterative scheme is implemented to coordinate the control inputs of the network: once all agents have solved their optimization problems for optimal control inputs, they communicate their solutions to the agents with whom they cooperate and, subsequently, proceed solving their optimization problems with the updated solutions. This process is repeated for $n_{\text{alt}}$ iterations.

B. Operational constraints

As previously discussed, the fluctuations in metering rates are penalized by a term in the objective functions. Similarly, it is necessary to avoid fluctuations in the VSLs to improve driver safety and comfort [22], [23]. This is done with hard constraints because the VSLs are discrete decision variables. Hence, the hard constraints reduce the size of the solution space of the VSLs and, therefore, reduce the complexity of the optimization problems.

Two types of constraints on the VSLs are considered. The first constraint allows the VSLs to maximally change $\eta_k$ per controller sample:

$$
|U_{\text{VSL}}^i(k_c) - U_{\text{VSL}}^i(k_c+1)| \leq \eta_k, \quad \forall i \in I_{\text{VSL}}.
$$

(13)

The second constraint allows the VSLs that are on two consecutive freeway segments to maximally differ $\eta_k$ from each other:

$$
|U_{\text{VSL}}^i(k_c) - U_{\text{VSL}}^{i+1}(k_c)| \leq \eta_k, \quad \forall i : i,i+1 \subseteq I_{\text{VSL}}.
$$

(14)

C. Alternating optimization scheme

The agents use an alternating optimization scheme [12] to decompose the MINLP problems into two sub-problems: nonlinear non-convex continuous optimization problems to find RM rates, and discrete optimization problems to find VSL values. In the continuous optimization problems, the VSLs are constant variables, and vice versa, in the discrete optimization problems, the RM rates are constant variables. The optimization problems are solved one after another for $n_{\text{alt}}$ iterations. In every new iteration, the constant variables are updated with the control inputs that have been found in the previous iteration.

The continuous optimization problem for agent $s$ to find the optimal metering rates $\hat{U}_s^l(k_c)$ in its subsystem at time $k_c$ over the prediction horizon $N_p$ is formulated as:

$$
\min_{U_s^l(k_c)} J_{\text{con}}^s(k_c)
$$

subject to:

$$
\begin{align*}
&x(k_c + l + 1) = f_M((x(k_c + l), \hat{U}_s^l(k_c + l)), \hat{U}_s^l(k_c + l), \hat{U}_{\text{VSL}}^s(k_c + l), d(k_c + l)), \\
x(k_c) = x_k, \\
d(k_c + l) = d_k, \\
U_s^l(k_c + l) \in \mathcal{U}_s, \forall i \in I_s^c, \\
U_s^l(k_c + j) = U_s^l(k_c + N_s - 1), \forall i \in I_s^c, \\
\text{for } j \in \{N_s, \ldots, N_p - 1\}, \\
\text{for } l \in \{0, 1, \ldots, N_p - 1\},
\end{align*}
$$

(16)

where $J_{\text{con}}^s$ is defined in (11) for FC-A-MPC and in (12) for DC-A-MPC, $\hat{U}_s^l$ and $\hat{U}_{\text{VSL}}^s$ contain all the RM rates and VSLs.
that are constant variables in the optimization, the future states $x$ of the network are predicted by the system dynamics $f_M$ of METANET. $x_d$ are the measurements of the states at time $k_c$, $d$ are the demands in the network, $d_k$ are the measurements of the demands at time $k_c$, $U_t^i$ is the input space of the RM rates, and $N_u$ is the control horizon that simplifies the optimization problem.

Similarly, the discrete optimization problem for agent $s$ to find the optimal speed limits $\bar{U}_{VSL}(k_c)$ in its subsystem at time $k_c$ over the prediction horizon $N_p$ is formulated as:

$$\min_{\bar{U}_{VSL}(k_c)} J_{\text{dis}}^s(k_c)$$

subject to:

$$x(k_c + l + 1) = f_M\left(x(k_c + l), \bar{U}_{VSL}(k_c + l), U_t^s(k_c + l), d(k_c + l)\right),$$

$$x(k_c) = x_k,$$

$$d(k_c + l) = d_k,$$

$$\bar{U}_{VSL}(k_c + j) = U_{VSL}(k_c + j), \forall i \in I^s_{VSL},$$

$$\bar{U}_{VSL}(k_c + j) = U_{VSL}(k_c + j), \forall i \in I^s_{VSL},$$

for $j \in \{N_u, \ldots, N_p - 1\},$

for $l \in \{0, 1, \ldots, N_p - 1\},$$$

where $U_{VSL}$ is the input space of the VSLs and $J_{\text{dis}}^s$ is defined in (11) for FC-A-MPC and in (12) for DC-A-MPC.

D. Initialization and stopping criteria

To initialize the iterative scheme, values for the constant variables in the optimization problems of the first distributed iteration are needed. Therefore, the agents use the time-shifted VSLs of the previous controller sample for these initial values [12].

The agents use two stopping criteria to terminate the optimization scheme:

1) $\text{CT} \geq t_{\text{term}}$: The Computational Time (CT) of the agents becomes larger than $t_{\text{term}}$.

2) $n_{\text{dist}}$: The optimization problems of the final distributed iteration $n_{\text{dist}}$ are solved.

Algorithm 1: Top level communication and optimisation protocol of FC-A-MPC and DC-A-MPC for agent $s$.

Input: $U_{VSL}^{\text{prev}}, U_t^s$, $t_{\text{term}}, n_{\text{dist}}, n_{\text{alt}}$

Output: $U_t^s$, $U_{VSL}^s$

Start $\text{timer}$

Set initial values

$$\bar{U}_{VSL}^s := U_{VSL}^{\text{prev}},$$

$$\hat{U}_t^s := U_t^{\text{prev}}.$$ $

Start iterative scheme

while $\text{timer} \leq t_{\text{term}}$ do

for $\text{dist} = 1$: $n_{\text{dist}}$ do

for $\text{alt} = 1$: $n_{\text{alt}}$ do

Solve continuous optimisation for $\hat{U}_t^s$

Update local part of $\hat{U}_t^s$ with $\hat{U}_t^s$\n
Solve discrete optimisation for $\bar{U}_{VSL}^s$

Update local part of $\bar{U}_{VSL}^s$ with $\bar{U}_{VSL}^s$

end

Store $\hat{U}_t^s$ and $\bar{U}_{VSL}^s$

Communicate $\hat{U}_t^s$ and $\bar{U}_{VSL}^s$ to other agents

Receive optimal inputs $\hat{U}_t^{\text{other}}$ and $\bar{U}_{VSL}^{\text{other}}$ of other agents

Update nonlocal part of $\hat{U}_t^s$ with $\hat{U}_t^{\text{other}}$

Update nonlocal part of $\bar{U}_{VSL}^s$ with $\bar{U}_{VSL}^{\text{other}}$

end

end

E. Algorithm formulation

Both FC-A-MPC and DC-A-MPC use the same communication and optimization protocols, formulated in Algorithm 1 and illustrated in Fig. 1.

The constant variables $\hat{U}_t^s$ and $\bar{U}_{VSL}^s$ of agent $s$ contain the intermediate inputs of agent $s$ (local part) and the intermediate inputs of all the other agents with whom agent
s cooperates (nonlocal part). To initialize the algorithms, the control inputs of the previous controller sample \( U_{VSL}^{\text{prev}} \) and \( U_{VSL}^{\text{prev}} \) are used as constant variables. Every agent solves the continuous optimization and the discrete optimization \( r_{alt} \) times for every distributed iteration. The local part of \( U_{alt}^{s} \) is updated with \( U_{alt}^{s} \) every time agent \( s \) solves the continuous optimization. Similarly, the local part of \( U_{VSL}^{s} \) is updated with \( U_{VSL}^{s} \) every time agent \( s \) solves the discrete optimization. After \( r_{alt} \) iterations, the agents communicate their optimal inputs \( U_{alt}^{s} \) and \( U_{VSL}^{s} \) to the agents with whom they cooperate and similarly resolve the optimal inputs \( U_{\text{other}}^{s} \) and \( U_{VSL}^{\text{other}} \). Subsequently, all agents update the nonlocal parts of \( U_{alt}^{s} \) and \( U_{VSL}^{s} \) with \( U_{\text{other}}^{s} \) and \( U_{VSL}^{\text{other}} \), and resolve to the next distributed iteration.

The algorithms are terminated either after \( n_{alt} \) distributed iterations or when the CT is larger than \( t_{sim} \). After every distributed iteration, the local inputs that have been found are stored. Finally, the global objective function is evaluated for the inputs that have been found in every distributed iteration. The first time samples of the inputs that yield the best global objective function value are sent to the network as control inputs.

IV. CASE STUDY

In this section, the proposed algorithms are evaluated in a hypothetical case study. Firstly, the set-up and the traffic scenario of the case study are outlined. Secondly, performance criteria for the algorithms are defined. Then, the no-control network response is investigated. Subsequently, the control parameters of the algorithms are described. Then, the results are presented in two parts. Firstly, the influence on the system performance by relaxing the VSLs to be continuous decision variables in a distributed setting is shown. Secondly, a comparison is presented between the proposed distributed algorithms and the the more conventional centralized and decentralized algorithms.

A. Case set-up and scenario

The freeway network from [17] is used, but is slightly modified to include RM installations. The network, depicted in Fig. 3, has a length of 30 km, is partitioned into \( N_{all} = 24 \) segments in set \( I_{all} \), contains six VSLs on segments \( I_{VSL} = \{2, 3, 9, 10, 16, 17\} \), contains three RM installations at the on-ramps on segments \( I_{r} = \{7, 14, 21\} \) and contains three off-ramps at segments \( I_{alt} = \{5, 12, 19\} \). The VSLs are allowed to take values from the discrete set \( VSL_{\text{set}} = \{40, 60, 80, 100\} \) km/h. The control inputs are enumerated in the downstream direction to improve readability as illustrated in Fig. 3.

METANET is used to simulate this network, whose system dynamics are described in Section II-B. The system parameters of the network are chosen identically to [17]. Hence, the only model parameter that differs per segment is the segment length. The segments that contain VSLs have a larger length, so that the speed limits have more influence on the overall traffic state of the network. To avoid spillback to hypothetical urban roads, soft constraints are imposed on the queue lengths on all three on-ramps of \( w_{\text{max}} = 100 \) veh.

A hypothetical traffic scenario is simulated for 2.5 hours, which corresponds to \( N_{sim} = 900 \) samples with model sampling time \( T = 10 \) s. The demand profile and splitting fractions are chosen similarly to [17], but are modified so that larger traffic jams occur when no control is applied. The splitting fractions are constant during the simulations: \( \beta_{2}(k) = 0.21, \beta_{12}(k) = 0.26, \beta_{10}(k) = 0.02, \forall k \). The demand profile is shown in Fig. 2.

All the simulations are conducted on an HP ZBook Studio G4, containing an Intel Core i7 processor and 8GB of RAM. The simulations are evaluated in MATLAB R2018b.

Fig. 2. Demands at the mainstream and on-ramps.

B. Performance criteria

Two performance criteria are used to evaluate the MPC algorithms in terms of system performance and computational complexity.

1) System performance: The TTS of the comprehensive freeway network is used to quantify the system performance:

\[
TTS = \sum_{k=1}^{N_{sim}} \left( \sum_{i \in I_{r}} \rho_{i}(k) L_{i} + \sum_{i \in I_{alt}} w_{i}(k) \right).
\]

(19)

For convenience, this performance index is also expressed as a reduction TTS\(_{\text{red}}\) relative to the no-control case:

\[
TTS_{\text{red}} = \frac{TTS_{nc} - TTS}{TTS_{nc}} \cdot 100\%.
\]

(20)

where TTS\(_{nc}\) is the TTS of the no-control case.

2) Computational complexity: To quantify the computational complexity of the algorithms, the CT that is needed to determine the control inputs for every controller sample is investigated. For the CT of the distributed algorithms, a summation is made of the computational times for the number of distributed iterations. The largest CT of all controller samples is denoted as CT\(_{max}\). Hence, a control algorithm is implementable in real-time if CT\(_{max}\) is smaller than the controller sampling time \( T_{c} \).

C. No-control system response

In an uncontrolled setting, all ramps are open and all VSLs are equal to the maximum speed limits, such that \( U_{alt}^{r}(k) = 1, \forall i \in I_{r}, \forall k \) and \( U_{VSL}^{r}(k) = 100 \) km/h, \( \forall i \in I_{VSL}, \forall k \).

The consequence of not controlling the high traffic demand is a major traffic congestion in the network. Two large traffic jams occur that spread over a large part of the network, illustrated in Fig. 4. During the traffic jams, not all vehicles can freely enter the network from on-ramps. Hence, queues originate at all three on-ramps, illustrated in Fig. 5. The no-control system performance is TTS\(_{nc}\) = 5986 veh-h.
The following MPC parameters are used in the case study.

1) System partitioning: The network is partitioned into three subsystems for the decentralized and distributed algorithms. It is partitioned in such a way that each subsystem contains one RM installation and two VSLs: \( I_1 = \{1, 2, \ldots, 7\}, I_2 = \{8, 9, \ldots, 14\} \) and \( I_3 = \{15, 16, \ldots, 24\} \). Previous works have shown that the alternating optimization can be solved accurately with this combination of actuators [12].

2) Sampling time and horizons: All algorithms in this work use a sampling time of \( T_c = 1200 \) s. To match this sampling time, the algorithms use \( N_p = 10 \). Hence, the network response of 1200 s in the future is taken into account. Consequently, at free-flow speed, the algorithms take into account the influences of vehicles over a spatial distance of 34 km, which is slightly larger than the network size. To simplify the optimization problems, \( N_u = 3 \) is used.

3) Number of alternating iterations: The decentralized and distributed algorithms use \( n_{\text{alt}} = 2 \) because their performance converges within two alternating iterations. The centralized algorithm requires \( n_{\text{alt}} = 5 \) because it has significantly larger optimization problems.

4) Weighting terms: The term \( \zeta \) is chosen zero to avoid a tedious tuning process and to make the comparison of the different algorithms more straightforward. If less fluctuation in the RM signals is desirable, \( \zeta \) can be increased.

The soft constraint term is chosen \( \zeta_w = 10 \), as this results in a good trade-off between queue length behavior and system performance. A 10% violation in queue constraints is considered acceptable in this work, since a hypothetical network is used without details on the urban roads surrounding the network. If less violation is desirable, \( \zeta_w \) can be increased.

The operational constraints are chosen \( \eta_l = \eta_u = 20 \) km/h, similarly to [12].

5) Optimization algorithms: All the continuous optimization problems are solved with Sequential Quadratic Programming (SQP) [24] combined with a multi-start approach. This approach is well known to properly solve non-convex nonlinear optimization problems and has shown good results in other works [5], [12].

The discrete optimization problems of the decentralized and distributed algorithms are solved by evaluating all the feasible solutions. Hence, this results in the global optimum of the subproblem. The size of the discrete solution space, and therefore the time needed to evaluate all the feasible solutions, increases exponentially with the size of the optimization problem. Due to this exponential growth, it is not possible to evaluate all the feasible solutions with the centralized algorithm in reasonable time. Therefore, a Genetic Algorithm (GA) [24] is used to solve the discrete optimization problems of the centralized algorithm.

The Optimization Toolbox in MATLAB is used to perform the SQP and GA optimization methods. The default SQP options of function \texttt{fmincon} are used, where \texttt{MaxFunctionEvaluations} and \texttt{MaxIterations} are both modified to \( 1 \cdot 10^7 \). The default GA options of function \texttt{ga} are used, where \texttt{PopulationSize} is modified to 800 and \texttt{MaxStallGenerations} is modified to 400.

E. Results: relaxation on VSLs

The performance of the proposed algorithms has been compared to the performance of the Fully Cooperative Rounding Model Predictive Control (FC-R-MPC) [5] and Downstream Cooperative Rounding Model Predictive Control (DC-R-MPC) algorithms. Both these algorithms have the same distributed architectures as the proposed algorithms, but consider the VSLs as continuous decision variables and afterwards round these to acceptable values.

Two different cases have been investigated: the first case only considers one distributed iteration \( n_{\text{dist}} = 1 \) and the second case considers multiple distributed iterations \( n_{\text{dist}} = 4 \). In both cases, the termination time is set to \( t_{\text{term}} = \infty \), so that the convergence of the system performance for increasing \( n_{\text{dist}} \)
can be compared.

The results are summarized in Table I and II. In the worst case, the relaxation results in a performance loss of 4.42\%, corresponding to the fully cooperative architecture with one distributed iteration, where the scheme that uses alternating optimization results in $TTS_{\text{red}} = 25.14\%$, while the scheme that uses rounding results in $TTS_{\text{red}} = 21.83\%$. Moreover, the schemes that use alternating optimization find solutions much closer to the optimum with the first distributed iteration.

Hence, relaxing the VSLs to be continuous decision variables results in a significant performance loss with a distributed architecture. This confirms the results in [12] and extends them to the distributed case.

### Table I

<table>
<thead>
<tr>
<th>Controller</th>
<th>TTS [veh/h]</th>
<th>$TTS_{\text{red}}$ [%]</th>
<th>$n_{\text{dist}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC-A-MPC</td>
<td>4515</td>
<td>24.57</td>
<td>1</td>
</tr>
<tr>
<td>DC-A-MPC</td>
<td>4505</td>
<td>24.74</td>
<td>4</td>
</tr>
<tr>
<td>DC-R-MPC</td>
<td>4685</td>
<td>21.73</td>
<td>1</td>
</tr>
<tr>
<td>DC-R-MPC</td>
<td>4612</td>
<td>22.95</td>
<td>4</td>
</tr>
</tbody>
</table>

### Table II

<table>
<thead>
<tr>
<th>Controller</th>
<th>TTS [veh/h]</th>
<th>$TTS_{\text{red}}$ [%]</th>
<th>$n_{\text{dist}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FC-A-MPC</td>
<td>4481</td>
<td>25.14</td>
<td>1</td>
</tr>
<tr>
<td>FC-A-MPC</td>
<td>4466</td>
<td>25.39</td>
<td>4</td>
</tr>
<tr>
<td>FC-R-MPC</td>
<td>4679</td>
<td>21.83</td>
<td>1</td>
</tr>
<tr>
<td>FC-R-MPC</td>
<td>4587</td>
<td>23.37</td>
<td>4</td>
</tr>
</tbody>
</table>

### F. Results: architecture comparison

The performance of the proposed distributed algorithms has been compared to the Centralized Alternating Model Predictive Control (Cent-A-MPC) and Decentralized Alternating Model Predictive Control (Dec-A-MPC) algorithms. Both algorithms also use an alternating optimization scheme, but have a centralized or decentralized architecture, respectively. The stopping criteria of the proposed algorithms have been set to $t_{\text{term}} = T_c$ and $n_{\text{dist}} = \infty$, such that they are implementable in real-time.

The performance of the four algorithms and the uncontrolled system is summarized in Table III. As expected, the centralized algorithm achieves the best system performance ($TTS_{\text{red}} = 25.63\%$). However, because only one agent controls the network by using one comprehensive system model, the computational complexity is exceptionally high. As a result, the algorithm is far from implementable because the optimization problems cannot be solved within the controller sampling intervals. On the other hand, the decentralized algorithm has, as expected, the lowest computational complexity because three agents are controlling the network without cooperation. However, due to the lack of cooperation, the system performance is also rather sub-optimal ($TTS_{\text{red}} = 20.85\%$).

The proposed algorithms offer a trade-off between computational complexity and system performance. The computational complexities of the proposed algorithms are significantly lower than the computational complexity of the centralized algorithm because three agents are controlling the network. As a consequence, the proposed algorithms manage to solve the optimization problems within the controller sampling intervals. Moreover, both have a significantly better system performance than the decentralized algorithm because the agents are actively cooperating to improve the global network performance.

The system performance of FC-A-MPC ($TTS_{\text{red}} = 25.44\%$) is similar to the optimal performance of the centralized algorithm. Both manage to keep the network mostly uncongested, illustrated in Fig. 6. There is a critical point in both simulations where all three on-ramps are close to the soft constraint of $w_{\text{max}} = 100$ veh, illustrated in Fig. 7. With both algorithms, the agents decide that slightly violating the soft constraint at the second on-ramp is beneficial for the network performance. However, the queue lengths remain within 10\% of the soft constraint and the violations can be avoided if necessary by increasing $\zeta_w$.

DC-A-MPC results in a worse system performance ($TTS_{\text{red}} = 24.74\%$) than FC-A-MPC because the agents are not fully cooperating. However, the computational complexity of DC-A-MPC is also lower because it only scales with the size of the subsystems.

Both the centralized algorithm and FC-A-MPC are actively limiting vehicles at the first two on-ramps and are using the first two VSLs to avoid a traffic jam in front of the third on-ramp, illustrated in Fig. 9. On the other hand, with DC-A-MPC, the first agent is barely limiting vehicles from entering the bottleneck at the third on-ramp, but the second agent is actively limiting the throughput to the bottleneck. As a result, a traffic jam originates in front of the third on-ramp. Lastly, with the decentralized algorithm, the first two agents are keeping their ramps open during the whole simulation due to the complete absence of cooperation. The result is a major traffic jam in front of the third on-ramp. Consequently, the queue at the third on-ramp violates the soft constraint significantly. The outflows and densities of the bottleneck at segment 21 for all algorithms are illustrated in Fig. 8.
Fig. 6. Heat maps of the densities for the different algorithms.

Fig. 7. Queue lengths in front of the on-ramps for the different algorithms.

Fig. 8. The outflows of and densities on segment 21 for the different algorithms.
In this paper, two similar MPC algorithms that use a distributed control architecture and an alternating optimization scheme for coordinated control of discrete VSLs and continuous RM rates on large freeway networks have been proposed. The algorithms have been evaluated in a case study.

Firstly, the effects on the system performance by relaxing the VSLs to be continuous decision variables have been investigated. To the best of the authors knowledge, this is the first time that these effects have been investigated with a distributed architecture. In the worst case (with a fully cooperative architecture and one distributed iteration), the relaxation results in a performance loss of 4.42%. Hence, the relaxation results in a significant performance loss with a distributed architecture. This extends the findings in [12] to the distributed case. Therefore, the contradictory conclusions in the field on this relaxation have been disproven more strongly now.

Secondly, the performance of the proposed algorithms has been compared to the performance of the more conventional centralized and decentralized algorithms in terms of system performance and computational complexity. The proposed algorithms offer a trade-off between computational complexity and system performance. Both have a significantly lower computational complexity than the centralized algorithm, since they do manage to solve all optimization problems within the controller sampling intervals. Moreover, the performance of FC-A-MPC is remarkably similar to the optimal performance of the centralized algorithm (TTS$_{red} = 25.44\%$ and TTS$_{red} = 25.63\%$, respectively).

It highly depends on the size of the freeway network which algorithm is more suitable. For the network in the case study, FC-A-MPC has the best overall performance, as it has the best system performance of the algorithms that manage to solve all their optimization problems within the controller sampling intervals. However, for even larger freeway networks, FC-A-MPC might computationally become too complex, while DC-A-MPC could still be implementable.

V. CONCLUSION

REFERENCES

José Ramón D. Frejo received the Ph.D. degree in automatic, robotic, and telematics from the University of Seville in 2015. He was an Assistant Professor with the University of Seville from 2009 to 2010, 2014, and 2015 to 2017, and with Universidad Loyola Andalucia from 2014 to 2015. He was a Visiting Scholar with the University of California at Berkeley, Berkeley, in 2011, with the Delft University of Technology, The Netherlands, in 2012, and with the Technical University of Crete in 2013 and 2017. Since 2017, he has been an MSCA-IF Research Fellow with the Delft Center for Systems and Control, Delft University of Technology. His research interests lie in the areas of model predictive control and freeway traffic control. He received the Outstanding Doctorate Award from the University of Seville in 2015 and the first prize of the IEEE ITS Best Dissertation Award in 2016.

Bart De Schutter received the Ph.D. degree (summa cum laude) in applied sciences from Katholieke Universiteit Leuven, Leuven, Belgium, in 1996. He is currently a Full Professor with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands. His current research interests include the control of intelligent transportation systems, hybrid and discrete-event systems, and multiagent systems. He is a Senior Editor of IEEE Transactions on Intelligent Transportations Systems.

Uglješa Todorović received the B.Sc. degree (cum laude) in mechanical engineering from Delft University of Technology, Delft, The Netherlands in 2016. He is currently finishing the M.Sc. degree in systems and control at Delft University of Technology, Delft, The Netherlands, focusing on distributed model predictive control applied to freeway networks.


Master of Science Thesis

Uglješa Todorović


Glossary

List of Acronyms

**Cent-A-MPC**  Centralised Alternating Model Predictive Control
**CT**  Computational Time
**CTM**  Cell Transmission Model
**DC-A-MPC**  Downstream Cooperative Alternating Model Predictive Control
**DC-R-MPC**  Downstream Cooperative Rounding Model Predictive Control
**Dec-A-MPC**  Decentralised Alternating Model Predictive Control
**FC-A-MPC**  Fully Cooperative Alternating Model Predictive Control
**FC-R-MPC**  Fully Cooperative Rounding Model Predictive Control
**FD**  Fundamental Diagram
**GA**  Genetic Algorithm
**MILP**  Mixed Integer Linear Programming
**MINLP**  Mixed Integer Nonlinear Programming
**MLD**  Mixed Logical Dynamical
**MPC**  Model Predictive Control
**RM**  Ramp Metering
**SQP**  Sequential Quadratic Programming
**TTS**  Total Time Spent
**VSL**  Variable Speed Limit
List of Symbols

Symbols related to METANET

- $k$: sample counter
- $T_m$: model sampling time [h]
- $\rho_i(k)$: density of segment $i$ at time $k$ [veh/km/lane]
- $v_i(k)$: mean speed of segment $i$ at time $k$ [km/h]
- $q_i(k)$: flow leaving segment $i$ at time $k$ [veh/h]
- $V(\rho_i(k))$: desired speed of drivers on segment $i$ at time $k$ [km/h]
- $w_i(k)$: length of the queue at the on-ramp of segment $i$ at time $k$ [veh]
- $d_i(k)$: traffic demand at the on-ramp on segment $i$ at time $k$ [veh/h]
- $U_i^r(k)$: ramp metering rate at the on-ramp at segment $i$ at time $k$
- $U_i^{VSL}(k)$: variable speed limit on segment $i$ at time $k$
- $\beta_i(k)$: fraction of flow that leaves segment $i$ from an off-ramp at time $k$
- $\lambda$: number of lanes in the freeway network
- $L_i$: length of segment $i$ [km]
- $\tau$: time constant of the speed relaxation term [h]
- $\kappa$: speed anticipation parameter [veh/km/lane]
- $\eta_l$: speed anticipation term for when a downstream density is lower than the density in the actual segment [km$^2$/h]
- $\eta_h$: speed anticipation term for when a downstream density is higher than the density in the actual segment [km$^2$/h]
- $a$: parameter of the fundamental diagram
- $v_{free}$: free-flow speed [km/h]
- $\rho_{max}$: the maximum density of a segment [veh/km/lane]
- $\rho_{crit}$: the critical density of a segment [veh/km/lane]
- $C$: capacity of an on-ramp [veh/h]
- $\delta$: model parameter for speed drop caused by merging phenomena
- $\alpha$: model parameter expressing non-compliance of drivers
- $w_{max}$: soft constraint on the number of vehicles in a queue at an on-ramp
- $VSL_{set}$: set of prescribed values that the speed limits are allowed to take
- $N_{all}$: number of segments in the network
- $I_{all}$: set of segments of the network
- $I_r$: set of segments that contain a ramp metering installation
- $I_{VSL}$: set of segments that contain a variable speed limit
- $I_{off}$: set of segments that contain an off-ramp
Symbols related to MPC

\( k_c \) controller sample counter
\( T_c \) controller sampling time [h]
\( M \) integer that relates controller time step \( k_c \) and model time step \( k \)
\( x(k_c) \) states of the state-space system at time \( k_c \)
\( x_k \) measurement of the states at time \( k_c \)
\( d_k \) measurement of the demands at time \( k_c \)
\( J(k_c) \) objective function at time \( k_c \)
\( N_{\text{obj}} \) number of sub-objectives in the comprehensive objective function
\( N_p \) prediction horizon
\( N_u \) control horizon
\( N_{\text{sub}} \) number of subsystems
\( I_s \) set of segments in subsystem \( s \)
\( I_{sr} \) set of segments in subsystem \( s \) that contain a ramp metering installation
\( I_{VSL} \) set of segments in subsystem \( s \) that contain a variable speed limit
\( I_{\text{off}} \) set of segments in subsystem \( s \) that contain an off-ramp
\( n_{\text{dist}} \) number of distributed iterations
\( n_{\text{alt}} \) number of alternating iterations
\( t_{\text{term}} \) termination time of the proposed algorithms
\( \zeta_r \) weighting term in the objective functions to penalise fluctuations in metering rates
\( \zeta_w \) weighting term in the objective functions to include soft constraints on the queue lengths
\( \eta_t \) parameter describing hard constraints on the input space of the variable speed limits
\( \eta_{\text{dl}} \) parameter describing hard constraints on the input space of the variable speed limits
\( \hat{U}_{sr}(k_c) \) optimal metering rates for all the metering installations in subsystem \( s \) over the prediction horizon \( N_p \) at time \( k_c \)
\( \hat{U}_{VSLr}(k_c) \) optimal speed limits for all the matrix sign in subsystem \( s \) over the prediction horizon \( N_p \) at time \( k_c \)
\( \hat{U}_{sr}(k_c) \) all ramp metering rates that are considered constant in the optimisation of agent \( s \) at time \( k_c \)
\( \hat{U}_{VSLr}(k_c) \) all variable speed limits that are considered constant in the optimisation of agent \( s \) at time \( k_c \)
\( U_t \) input space of the ramp metering rates
\( U_{VSL} \) input space of the variable speed limits