Efficient Algorithms for Network-Wide Road Traffic Control

van de Weg, Goof Sterk

DOI
10.4233/uuid:dd6d52a5-b091-44c1-ba45-f96c8c3c3590

Publication date
2017

Document Version
Final published version

Citation (APA)
van de Weg, G. S. (2017). *Efficient Algorithms for Network-Wide Road Traffic Control*. [Dissertation (TU Delft), Delft University of Technology]. TRAIL Research School. https://doi.org/10.4233/uuid:dd6d52a5-b091-44c1-ba45-f96c8c3c3590

TRAIL THESIS SERIES

Goof Sterk van de Weg

Efficient Algorithms for Network-Wide Road Traffic Control

Efficient Algorithms for Network-Wide Road Traffic Control

Goof Sterk van de Weg

Delft University of Technology, 2017

This thesis is a result from a project (partly) funded by the Netherlands Organisation for Scientific Research (NWO), Delft University of Technology, and the Netherlands Research School on Transport, Infrastructure and Logistics (TRAIL).

Nederlandse Organisatie voor Wetenschappelijk Onderzoek

Cover illustration: Goof Sterk van de Weg and Theun Okkerse c/o Pictoright

Efficient Algorithms for Network-Wide Road Traffic Control

Proefschrift

ter verkrijging van de graad van doctor

aan de Technische Universiteit Delft,

op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,

voorzitter van het College voor Promoties,

in het openbaar te verdedigen op donderdag 26 oktober 2017 om 10.00 uur

door

Goof Sterk VAN DE WEG

Master of Science in Systems and Control

Bachelor of Science in Mechanical Engineering

Delft University of Technology (the Netherlands)

geboren te Dordrecht, Nederland

This dissertation has been approved by the
promotor: Prof. dr. ir. S.P. Hoogendoorn
copromotor: Dr. ir. A. Hegyi

Composition of the doctoral committee :

Rector Magnificus	Chairman
Prof. dr. ir. S.P. Hoogendoorn	Technische Universiteit Delft
Dr. ir. A. Hegyi	Technische Universiteit Delft

Independent members:

Prof. dr. M. Menéndez	ETH Zürich, Switzerland
Prof. dr. ir. I.J.B.F. Adan	Technische Universiteit Eindhoven
Prof. dr. H. Vu	Monash Univeristy, Australia
Prof. dr. ir. J.W.C. van Lint	Technische Universiteit Delft
Prof. dr. ir. B. De Schutter	Technische Universiteit Delft

This thesis is the result of a Ph.D. study carried out from 2013 to 2017 at Delft University of Technology, Faculty of Civil Engineering and Geosciences, Transport and Planning Section.

TRAIL Thesis Series no. T2017/11, the Netherlands TRAIL Research School

TRAIL
P.O. Box 5017
2600 GA Delft
The Netherlands
E-mail: info@rsTRAIL.nl

ISBN 978-90-5584-229-2

Printed in The Netherlands

Yesterday I woke up sucking a lemon
Everything in its right place
- Radiohead (*1999)*

Voorwoord

Een proefschrift is meer dan alleen een bewijs van Stoïcijnse verdieping in een vakgebied, het is ook een bewijs van toewijding en persoonlijke ontwikkeling. Naast de unieke kans om zonder beperking maanden lang diep na te denken over een onderwerp, heeft het promotie traject me dan ook vele bijzondere ervaringen geboden op professioneel en persoonlijk vlak. Ik zou graag mijn dankbaarheid willen uitten aan een ieder die direct of indirect heeft bijgedragen aan deze periode en de totstandkoming van dit proefschrift .

Ten eerste wil ik Andreas Hegyi bedanken die me als afstudeerbegeleider heeft gemotiveerd om te gaan promoveren, wat me op dat moment de grootste uitdaging leek die ik kon aangaan. Ik ben je erg dankbaar voor het vertrouwen dat je altijd gehad hebt in een goede uitkomst en voor de vele wijze lessen. Zonder je kritische vragen over de ideeën die ik geregeld met je besprak hadden er nu niet 5 mooie algoritmes in dit proefschrift gestaan.

Serge Hoogendoorn wil ik bedanken voor het mogelijk maken van het promotie traject en voor de altijd scherpe kritiek en nieuwe invalshoeken. Daarnaast ben ik je ook dankbaar voor het voortdurende vertrouwen in mijn onderzoekspraktijken. Een goed voorbeeld hiervan is een bespreking in maart 2014. Omstreeks januari 2014 was ik me volledig gaan richten op stadsverkeersregelingen. Na 2 maanden diep nadenken vroeg ik me in deze bespreking af of de ideeën die ik had wel ergens naartoe leiden. De feedback die je gaf luidde ongeveer: “je hebt best aardige ideeën en ik vertrouw erop dat er iets goeds uitkomt, ga nog een maand zo door en als het dan niet duidelijker wordt kijken we samen hoe we het gerichter kunnen maken.” Gedurende die maand kwam inderdaad een ‘Eureka-moment’ en legde ik de basis voor het algoritme beschreven in hoofdstuk 4 van dit proefschrift.

Bart De Schutter wil ik graag bedanken voor de technische hulp bij de formulering van de optimalisatie aanpakken in de artikelen in hoofdstukken 3 en 5 en voor de tijdige en hoge kwaliteit feedback op de tekst van deze artikelen. Daarnaast wil ik je bedanken voor de leerzame samenwerking in de begeleiding van Dik Jansen tijdens zijn afstuderen. Ook wil ik je bedanken voor het plaatsnemen in de commissie.

I also want to thank Hai Le Vu for hosting my visit at the Swinburne University of Technology. Without our discussions and your help, I would not have been able to develop and implement the algorithm presented in Chapter 6. Besides that, I want

to thank you for the collaboration in supervising Dik Jansen during his visit at your group, and for taking place in the doctoral committee.

Thanks to the other doctoral committee members, namely, Ivo Adan, Hans van Lint, and Mónica Menéndez, for their valuable time invested in reading and providing feedback on the dissertation, and taking place in the doctoral committee.

A warm thanks to all the colleagues of the Transport & Planning department who have made the work so much more pleasant. Especially: Mario, Giselle, Tamara, Erika, Femke, Ramon, Alex, Pablo, Bernat, Meng, Yufei, Oded, Victor, Bart Wiegmans, and Nikola. Henk Taale, dank voor de hulp met hoofdstuk 5. Niharika, it was great working with you. Paul van Erp, dank voor de gezelligheid, het luisterend oor, en de vele ingewikkelde discussies over allerlei onderwerpen. Mehdi, I am very thankful for your help, it was lots of fun working with you! Edwin, dank voor de ondersteuning en hulp bij technische problemen. Priscilla en Dehlaila, bedankt voor de gezelligheid en de hulp met de verscheidene verzoeken waar jullie altijd snel mee aan de slag gingen, ik bewonder jullie talent om zoveel verschillende dingen tegelijk te doen.

Ook dank aan de afstudeerders Robin, Mark, Emiel, Niharika, Rien, en Dik die ik heb mogen begeleiden, voor de leerzame ervaring. In het bijzonder Dik, het was gezellig met je samen te werken, zeker ook in Melbourne. Ik ben erg trots op het goede resultaat dat je gehaald hebt!

Jaap, Gerard, Koen, dank voor de leuke en leerzame tijd bij Arane! Het was erg interessant om de praktische kant van de verkeersregelingen te zien en ik heb ook genoten van de goede sfeer in jullie team.

Theun, hartelijk bedankt voor het ontwerpen van de omslag en de figuren in de introductie. Ik vind het fantastisch hoe je mijn interesses kunstzinnige hebt weten te verbinden met de inhoud van dit proefschrift, dank je wel!

Het is erg gemakkelijk om elke minuut van de dag bezig te zijn met een promotie onderzoek. Gelukkig heb ik kunnen rekenen op een heel stel vrienden, hoewel het me niet altijd gelukt is om niet over werk te praten. Ik denk dat ik er wel steeds beter in word, gelukkig. Job, wat hebben we veel mooie avonturen beleefd, dat zullen we vast nog wel doorzetten! Freek, ik vind het altijd fantastisch om weer bij te praten, alsof de tijd heeft stil gestaan. Bart-Jan en Rik, het was een mooie studententijd met jullie en leuk om daar nog van na te genieten in Den Haag. Jan, Elise, Floris, Thijs, Koen en Luuk, heel tof om nog altijd bij elkaar te komen en hopelijk gaan we nog eens op een reisje. Natuurlijk wil ik ook even de Zorbanen noemen; in het bijzonder Rein voor je wijsheid en gezelligheid, Stijn voor je enorme inspirerende energie en enthousiasme, en Philip voor je vriendschap waar ik altijd op kan rekenen. Erik-Sander het was super dat je ook op de afdeling werkte en hebt geholpen met hoofdstuk 6. Hopelijk zie ik jou, Emma en Bram nog vaak en gaan we snel weer wielrennen.

Dank aan Milou, Sander en Loek. Milou, dank voor de altijd open deur, de klus projecten waar ik me heerlijk heb op kunnen uitleven, het bbq'en, het vervoer op vakantie, en natuurlijk voor de bierbrouw hobby.

Tante Magda Thoeng wil ik bedanken voor haar liefde en zorgen, het is nog altijd heerlijk om u op te zoeken.

In memoriam wil ik ook nog even terug denken aan oma en opa van de Weg, oma en opa Voordouw en tante Jeanne en Oom Ben wiens liefde en voorbeelden mij hebben gemaakt tot wie ik nu ben. Opa Voordouw deelde mijn passie voor de techniek. Oom Ben was een groot voorbeeld van liefde, rust, eerbied, en moraliteit.

Cox, wijze zus, ik hoop dat we net zo veel bij elkaar over de vloer komen als de laatste jaren. Dank aan jou en Pascal voor alle steun en raad en heel veel geluk met elkaar, Pepijn en Tobias. Geert, grote broer, jij zorgt er altijd voor dat er wat meer avontuur is en ik bewonder je passie en toewijding voor de vele projecten die je mooi vindt. Christianne, lieve zus, dank voor alle gezelligheid en steun de afgelopen jaren en de lekkere, vernieuwende maaltijden en drankjes die je ons voorschotelt.

Lieve mama, dank voor alle liefde en steun en de open armen waarmee je ons altijd onthaald en natuurlijk de gevulde armen waar we dan weer mee weggaan. Cees ook bedankt voor de gezelligheid en steun.

Lieve papa, dank voor de liefde en warmte, de vele steun en wijze raad, en dat we altijd welkom zijn en op je kunnen rekenen.

Merle, liefste, ik ben je heel erg dankbaar voor al je steun en liefde. Je hebt denk ik het beste meegemaakt hoe ik deze 4 jaar heb doorstaan en zonder jou was het een stuk zwaarder geweest. We hebben super veel leuke dingen meegemaakt, en ik kijk uit naar alle leuke dingen die we nog samen gaan beleven.

Goof Sterk van de Weg
Rotterdam, 28 september 2017

Contents

Chapter 1

Introduction

Road traffic networks are not always utilized to their maximum potential. This means that travelers experience unnecessary delays due to, for instance, freeway congestion or inefficient use of intersections. These delays cause economical and societal costs. According to the European Commission congestion costs in Europe mount up to 1% of the gross domestic product [European Commission, 2014]. One important cause of this problem is the lack of efficient network-wide traffic control measures which is the main topic of this dissertation. More specifically, this dissertation focuses on feedback traffic control algorithms.

Feedback traffic control aims at influencing the traffic using actuators – for instance, variable speed limits (VSLs), ramp metering (RM), traffic lights, and route guidance – based on real-time traffic measurements [Papageorgiou et al., 2003]. A well-known example is ramp metering which is commonly used to limit an on-ramp flow using a traffic light at the on-ramp so that the freeway flow remains below the capacity of a bottleneck [Papageorgiou et al., 1988]. This causes a congestion reduction which is beneficial for the freeway performance because it reduces the impact of the capacity drop – i.e., the phenomenon that the congestion outflow is less than the free flow capacity [Hall and Agyemang-Duah, 1991, Kerner and Rehborn, 1996, Leclercq et al., 2016]. In this way, the average travel time of all the road-users is reduced because the freeway outflow remains higher.

Network-wide traffic control can be used to reach various objectives. Examples of these objectives are improving throughput, reducing pollution, improving safety, improving reliability, and improving equity [Burger et al., 2013]. The main objective for applying ramp metering in the example above is to improve the throughput. Besides that, reducing congestion may also lead to a reduction in pollution and in a safety gain. However, the improved throughput is realized by solely delaying the on-ramp traffic which may not always be equitable. Hence, different objectives may be conflicting. The task of a control algorithm is to influence the traffic so that the performance – expressed via one or a set of objectives – is improved. This dissertation focuses on the objective of improving the network-wide throughput.

The currently available traffic control algorithms are not always able to efficiently utilize the network capacity. One of the main reasons for this is that improving the network-wide throughput requires to coordinate numerous actuators throughout the network. Such coordination is theoretically challenging due to, among others, the complexity of traffic dynamics. Apart from that, coordinating a lot of actuators introduces a lot of decision variables which may cost a lot of computation time to optimize. As a consequence, most traffic control algorithms contain significant simplifications so that they require only a limited amount of computation time which often leads to sub-optimal performance.

Hence, there exists room for improving the currently available traffic control algorithms. In fact, there are several aspects on which traffic control algorithms can be improved. However, for the sake of simplicity, this dissertation investigates how traffic control algorithms can be developed that lead to a better balance between the following two requirements:

- The algorithm has to be able to *coordinate* multiple (different) actuators in order to maximize the network performance,
- The algorithm has to be *real-time feasible*. This means that it has to be able to compute the control signal within the controller sampling time which is typically in the range of one to several minutes.

It must be noted that these two requirements are often conflicting because the problem complexity typically increases when the number of actuators that need to be coordinated increases. As a consequence, an increase in the number of actuators may cause an increase in the computation time used by the control algorithm which may conflict with the real-time feasibility requirement. Therefore, it may be needed to realize a better balance between the realized network performance and the required computation time.

The aim of this dissertation is to design traffic control algorithms for network-wide traffic control that lead to a better balance between network performance and computation time. Before formalizing this aim into a research objective, first the background of the problem is introduced. To this end, the next section discusses the main characteristics of the network-wide traffic control problem. Section 1.2 details the main challenges and opportunities relevant for the design of network-wide traffic control measures in this dissertation. Section 1.3 then presents the research objective, followed by the research scope Section 1.4, and the research approach Section 1.5. Section 1.6 presents the main contributions and Section 1.7 presents the dissertation outline.

1.1 Problem characteristics

A network-wide traffic control system consists of detectors, actuators, state estimation algorithms, and control algorithms that influence the traffic as illustrated in Figure 1.1.

All these elements have different characteristics that need to be accounted for when developing network-wide traffic control algorithms. This section first describes the characteristics of the traffic dynamics followed by the characteristics of the actuators relevant for this dissertation.

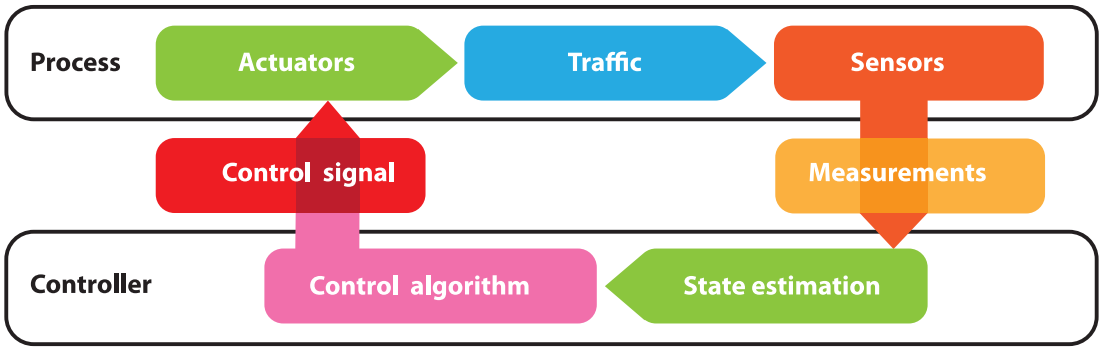


Figure 1.1: Overview of a traffic control system

1.1.1 Traffic dynamics

The propagation of traffic through a network is a dynamic process with many characteristics. Depending on the intended application of a traffic control algorithm it has to be able to account for several of these characteristics. Interestingly, the relevant characteristics of urban roads and freeways differ and as a consequence this section discusses these characteristics separately. Hence, this section first discusses the main characteristics of urban traffic dynamics and their implication on the design of traffic control algorithms, followed by a discussion of the characteristics of freeway traffic dynamics and their implication on the design of traffic control algorithms.

The traffic dynamics in an *urban* link can be divided into three traffic regimes. The division of the regime inside a link used in this dissertation is based on the definition presented by Aboudolas et al. [2010]. It must be noted though that in Aboudolas et al. [2010] a regime refers to the traffic situation inside the majority of the links in a network while in this dissertation it refers to the traffic situation inside individual links. The undersaturated regime represents the situation in which a queue can be emptied during a green time implying that a coupling from upstream to downstream intersections exists. In this regime, green waves can be created that allow vehicles to pass several intersections without stopping. The saturated regime is defined as the situation in which queues cannot be dissolved during a green time implying that no direct coupling between intersections exists. Green waves can no longer be created in this regime and the queue outflow equals the saturation rate if there is no downstream storage capacity limitation. The oversaturated regime is characterized by queues that propagate to upstream intersections causing a coupling from downstream intersections to upstream intersections. This coupling is time delayed, since, it takes time for the

space created by vehicles leaving the downstream intersection to reach the upstream intersection.

An urban traffic control algorithm has to account for different characteristics depending on the intended application. For instance, a traffic control algorithm designed for the undersaturated regime should be able to account for the downstream propagating waves caused by free flowing traffic. If this is not included, the controller will not be able to coordinate the off-set between intersections that is used to create green waves Little [1966], Little et al. [1981]. Similarly, if the upstream propagating waves caused by spillback are not accounted for by the control algorithm, the controller will tend to overestimate the remaining storage space in a link. Due to this, the controller may try to realize higher flows to a downstream link than physically possible while reducing the flows to other links resulting in a performance loss.

Several characteristics of *freeway* traffic dynamics are relevant for this dissertation. In free flow conditions the density – i.e., the number of vehicles in a link (or segment) – is positively correlated with the flow. In practice it is also observed that the speed in the link reduces when the density increases in free flow conditions. When the density reaches the critical density, traffic becomes unstable meaning that (small) disturbances may lead to congestion. Hence, the density and flow are negatively correlated for densities beyond the critical density. Congestion typically causes a capacity drop [Hall and Agyemang-Duah, 1991, Kerner and Rehborn, 1996, Leclercq et al., 2016]. Note that the capacity drop is usually not observed in urban traffic networks. The reason being that the maximum flows in urban traffic networks are realized by the outflows from queues that are already limited by the queue discharge rate. The severity of the capacity drop depends on several factors. One of these is the type of congestion. The two most well-known forms of congestion are jam waves – i.e., congestion with a length of roughly a few hundred meters to 2 km that propagate in the upstream direction – and standing queues. Typically, the capacity drop caused by a jam wave is larger (in the range of 30% according to Kerner and Rehborn [1996]) when compared to the capacity drop caused by a standing queue which is in the range of 10 to 13% according to Leclercq et al. [2016].

Similarly as for urban traffic, the intended application of a freeway traffic control algorithm influences the characteristics that need to be accounted for. In free flow conditions it is required to account for the travel times between different network elements. For instance, when coordinating the outflows of different on-ramps using RM to maximize the throughput of a downstream bottleneck, it may be beneficial to account for the time delay between the changes in the outflow of the upstream on-ramp onto the flow passing the downstream on-ramp and the bottleneck. Neglecting these free flow dynamics simplifies the control algorithm but may also introduce efficiency losses or controller instability. The capacity drop is an important property that is to be taken into account when developing traffic control algorithms for congested conditions. Not accounting for the capacity drop means that there is no difference between preventing or allowing congestion on a freeway stretch without off-ramps in terms of realized

freeway throughput. On the other hand, including the capacity drop may lead to a more complex controller design. Finally, a freeway traffic control algorithm designed for jam waves may not be efficient when applied to a standing queue and vice versa. However, developing an algorithm that is capable of accounting for both congestion types may be more complex.

1.1.2 Actuators used for network-wide traffic control

The actuators that are considered in a network-wide traffic control system have several characteristics that have to be considered as well. This dissertation is limited to four types of actuators, namely, traffic lights, (in-vehicle) variable speed limits, ramp metering installations, and route guidance. The characteristics of these actuators and the implication of these characteristics for the controller design are discussed below.

Traffic lights are a well-known and broadly used traffic control measure. Traffic lights at an intersection are controlled via a signal program, i.e., an algorithm that determines which streams can be active – i.e., is given a green light – at what time instant. A signal plan has several properties as will be detailed first [Hoornman and Bronkhorst, 2014, Papageorgiou et al., 2003]. A stage is a set of streams that can be active simultaneously. When the streams in two subsequent stages are conflicting, a clearance time has to be respected between the time when stopping one stage and activating the next in order to avoid collisions. In practice, a signal program consists of a fixed sequence of stages which may contain some degree of flexibility. A complete sequence of stages is referred to as a cycle. Typically, every stream receives a minimum amount of green time during a cycle and a maximum amount of green time in order to limit the maximum cycle time. Some signal plans use an offset between intersections. This offset enables the coordination of the signal programs of different intersections so that traffic leaving the upstream intersection receives a green light when reaching the downstream intersection. This is commonly known as the green wave [Little, 1966, Little et al., 1981]

These properties may affect the controller in several ways. Due to the clearance time, it is beneficial to increase the cycle time in the saturated and oversaturated regime. The reason for this is that a longer cycle time reduces the number of switches between stages which reduces the fraction of the cycle time that is not used by traffic. Despite the advantage of choosing a longer cycle time, it cannot be chosen too long, since, this may cause annoyance or, even worse, road users ignoring red lights. The sequence of stages can affect the performance as well. In practice, stage sequences are fixed. One of the main reasons for doing this is that road users get acquainted with the signal program so that changing the stage sequence may lead to confusion, annoyance or non-compliance. Another advantage of fixing the stage sequence is that it simplifies the control problem. On the other hand, fixing the sequence reduces the control freedom and as a consequence may reduce the performance. Finally, the off-set is commonly used for coordinating the signal plans of intersections in undersaturated regimes. This

concept may also be used in the oversaturated regime to coordinate the signal plans in the upstream direction.

Variable speed limits are commonly implemented using variable message signs (VMS) placed on gantries above a freeway and may also be displayed in the vehicle. While research has shown that VSLs can be used to improve the freeway throughput, they are typically used in practice to enhance the safety. An example is the automatic incident detection (AID) system used in the Netherlands. The AID system in the Netherlands displays a speed advice of 50 km/h if a speed below 50 km/h is detected by inductive loop detectors near the VMS gantry. Additionally, the gantry directly upstream of the gantry displaying 50 km/h displays a speed advice of 70 km/h. In this way, road users start limiting their speed and are aware that they are approaching congestion. According to Taale and Schuurman [2015] this system has led to an 18% reduction of head-to-tail collisions.

When applying a VSL system, the following characteristics should be included. First, a VSL controller has to be able to correctly account for the impact of the displayed VSLs on the traffic flow dynamics. According to Hegyi et al. [2010] two main approaches exist to improve the freeway throughput using VSLs. Homogenizing is the first approach which displays VSLs on VMS that are similar to the average speed of the traffic. This reduces the speed differences which stabilizes the traffic flow reducing the probability of traffic breakdown, and thus, leading to improved freeway throughput [Smulders, 1990, Van den Hoogen and Smulders, 1994, Kühne, 1991]. However, field-test results did not show significant throughput improvements [Van den Hoogen and Smulders, 1994]. Flow limitation is the second approach which aims at reducing the freeway flow by displaying VSLs. Field-test results using the SPECIALIST VSL algorithm showed that the flow into a jam wave can be reduced by displaying VSLs upstream of the jam wave [Hegyi et al., 2010]. Due to the flow reduction, the jam waves could be resolved leading to improved freeway throughput. Resolving a jam wave means that the upstream propagating high density, low speed state that characterizes a jam wave, is removed, so that it is possible to realize traffic flows up to the free flow capacity. Carlson et al. [2011] proposed an algorithm that applies VSLs upstream of a bottleneck so that the bottleneck inflow can be controlled to match the bottleneck capacity. This may prevent bottleneck congestion and maximize the throughput. Another property that has to be respected is compliance to the displayed speed limits. It is well known that the actual speed of traffic that is speed limited – also called the effective speed – is not equal to the displayed speed limits. Hence, a VSL controller has to account for the compliance of traffic to the VSLs. Finally, a VSL strategy should not cause unsafe situations, such as a situation where only a percentage of the road-users is speed-limited by VSLs or a situation where road-users experience sudden drops in the VSLs.

Ramp metering installations are traffic lights placed at on-ramps that allow a limited number of vehicles to enter the freeway when showing green. In this way, the freeway flow downstream of the on-ramp can be changed. One of the most well-known RM al-

gorithms is called ALINEA [Papageorgiou et al., 1988] and has been applied at several on-ramps throughout the world.

Several characteristics of RM installations have to be accounted for when developing a RM algorithm. The possible RM rates are bounded by a minimum and maximum RM rate. The minimum rate prevents excessive waiting while the maximum RM rate is a physical constraint caused by the minimum cycle time of the RM installation. The limitation of the on-ramp flow usually causes an on-ramp queue. Typically, this on-ramp queue has to be limited in order to avoid spillback to the upstream (urban) traffic network. The maximum queue length may limit the time over which RM can reduce the on-ramp flow and therewith limit its effectiveness.

Route guidance is a traffic control measure that can be used to re-route traffic. Route guidance can be realized using VMS by displaying routing advice at major bifurcations, or by displaying in-car messages, for instance, as part of a navigation system. One of the reasons for applying route guidance is to distribute traffic more efficiently over the different routes in a network [Papgeorgiou and Messmer, 1991]. Another reasons for implementing route guidance is to direct traffic away from incidents in the network.

Several characteristics of route guidance need to be considered when developing route guidance control algorithms. First, route guidance may cause an interaction effect between the road users and other traffic control measures. As an example, consider a system where road users have devices that decide based on the current traffic situation and potentially on the predicted travel times, what routes lead to the smallest travel time for the individual road user. When the control actions of other traffic control measures are not adapting to this re-routing effect, the network may get into a sub-optimal user optimum. Accounting for these influences requires an integrated control action that accounts for the impact of the infrastructure control actions onto the re-routing. However, coordinating the route choice with other control measures results in a complex problem. Second, people may not fully comply to the route guidance advice. Hence, a traffic control algorithm has to account for non-compliance or it should be combined with a policy that can realize a high compliance.

1.2 Challenges and opportunities of network-wide traffic control

Apart from the complexity introduced by the aforementioned characteristics that need to be accounted for, the main complicating factor of network-wide traffic control is (simply) the size of the network. Controlling the traffic in an urban region requires coordination of hundreds of traffic lights and actuators along many tens of kilometers of freeway. The number of control variables of such a system is enormous, causing computational issues. Besides that, developing algorithms to coordinate this number

of variables is also challenging from a theoretical point of view due to, for instance, the many problem characteristics that have to be considered.

A promising approach to control such networks is to divide the network into sub-networks. Such a *sub-network* is defined in this dissertation as a *medium-to-large scale network* consisting of tens of kilometers of freeway or tens of intersections. The sub-network controllers are then used to optimize the performance in the sub-networks while a higher level controllers optimizes the flows that are exchanged between the sub-networks leading to network-wide performance improvement. In this way, the sub-network controllers can consider more detail while the algorithm that has to coordinate the sub-network interaction can consider more simplified or aggregated dynamics. For instance, Hajiahmadi et al. [2015b] proposed a control strategy to coordinate the sub-network interaction based on the network fundamental diagram (NFD). Zhou et al. [2016] integrated that strategy in a hierarchical control framework as described above. The Rhodes algorithm is another example of a hierarchical control framework for urban traffic networks [Head et al., 1992].

This dissertation focuses on the design of algorithms for sub-networks in the light of a multi-level or hierarchical system as discussed above. Two types of sub-networks are considered, namely, freeway and urban sub-networks. This division is made, since, the characteristics of the problem of freeway and urban sub-networks are rather different so that different control designs are needed. Below, first the problems faced when developing freeway or urban traffic control algorithms are discussed. After that, Section 1.2.2 discusses opportunities for improving the algorithms. When needed, a distinction between freeway and urban traffic control is made.

1.2.1 Challenges

Ideally, a traffic control algorithm optimizes the control action of various actuators to maximize the throughput. Hence, various traffic control algorithms have been proposed in the scientific literature that are able to automatically select the control signals that optimize the network performance over a time horizon. See [Hegyi et al., 2005b, Gomes and Horowitz, 2006, Hajiahmadi et al., 2013a, Van den Berg et al., 2007] for optimization of the VSL or RM signals in freeway networks. See [Aboudolas et al., 2010, Le et al., 2013, Lin et al., 2012, Van den Berg et al., 2007] for optimization of the signal timings of intersection controllers to maximize the urban network throughput. Major advantages of these algorithms are that they can easily adjust to various traffic situations, various traffic demand patterns, and various road lay-outs while maintaining the ability to optimize the network performance. However, despite these advantages, this type of algorithm has not been implemented in practice due to several reasons. First, including all the relevant problem characteristics requires a complex optimization problem that does not always satisfy the real-time feasibility requirement. Second, optimizing the performance over a time-horizon requires a prediction of near-future traffic demands and turn-fractions at off-ramps and bifurcations, which is not readily

available. Third, the optimized control actions are not always insightful which affects the acceptance of the control strategies by the authorities.

In contrast to optimization-based algorithms, in practice mainly non-optimizing control algorithms of the feed-forward or feedback type are implemented that coordinate the control actions of a small number of actuators for a specific traffic situation. For examples of practice applied freeway traffic control algorithms see, [Papageorgiou et al., 1988] for feedback RM to prevent bottleneck congestion, [Middelham and Taale, 2006] for feed-forward RM, and [Hegyi et al., 2010] for a feed-forward VSL control algorithm. Examples of practice applied urban traffic control algorithms are, the TUC algorithm [Diakaki et al., 2003, Kraus Jr et al., 2010] which is a feed-back algorithm designed for the saturated regime, and SCOOT and SCATS which are algorithms designed for the undersaturated regime [Hunt et al., 1982, Luk, 1984]. The advantages of these algorithms are that they require little computation time, that they typically do not rely on demand predictions, and that they exploit simple or insightful algorithmic formulations. A disadvantage of these algorithms is that they may not be able to optimize the performance in all traffic situations. For instance, most urban traffic control algorithms do not consider the upstream propagating waves caused by spill back, while in that regime, strong relations between intersections exist, especially requiring coordination.

1.2.2 Opportunities of traffic control algorithm design

Recent technological innovations and scientific insights provide opportunities for improving both freeway and urban traffic control algorithms. Technological innovations can be used to provide better detection and actuation possibilities that may be used to improve the controller performance. Similarly, scientific insights may be used to develop new algorithms that make more efficient use of existing detection and actuation possibilities. In some cases, a combined approach may be followed in which new algorithms are developed that make efficient use of new detection and actuation possibilities.

The most relevant technological innovation for this dissertation is the rapid increase of in-vehicle technology, such as GPS navigation systems, enabling cooperative systems – i.e., systems in which vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communication is enabled. Due to this increase, the availability of floating car data, i.e., GPS speed and position data of individual road users, is increasing. This data is more detailed compared to the traffic data based on inductive loop detectors. For instance, Bayen and Patire [2010] showed in a field-test that estimates of the traffic state can be drastically improved by combining inductive detector loop data with FCD of just a few percent of the traffic. Hence, it has the potential to supply existing traffic control algorithms with more accurate traffic state estimations. Moreover, it may also be used to develop new traffic control algorithms that take the individual vehicle as the controlled element, instead of taking road segments as the controlled elements.

Apart from more accurate data, cooperative systems also provides new data types. A promising data type is the planned route of individual road-users. This information may be used to provide a better prediction of the future traffic demand.

Cooperative systems not only provide better data but can also be used to directly influence individual vehicles. This may be possible by displaying in-vehicle messages for instance on GPS navigation devices to re-route the traffic, provide speed advice to individual vehicles, or even by directly influencing the speed of individual vehicles. The advantage of this innovation is that it allows more detailed traffic control, since, the strict time-space discretization of control actions that is currently determined by the infrastructure can be relaxed. In order to benefit from this technology, new traffic control algorithms have to be designed that take the individual vehicle as the controlled element.

The most relevant insight for the development of *freeway* traffic control algorithms in this dissertation is the application of shock wave theory to describe the effect of VSLs on the traffic flow. These insights were used by the SPECIALIST algorithm that was capable of resolving jam waves on the A12 freeway in the Netherlands [Hegyi et al., 2010]. However, the SPECIALIST algorithm is a feed-forward algorithm designed for a conventional VSL system consisting of inductive detector loops and roadside VMS. Wang et al. [2014] showed that the use of cooperative systems can improve the performance of the SPECIALIST algorithm. However, in order to fully benefit from cooperative systems, a new algorithmic formulation may be needed that considers the individual vehicle as the controlled element. Apart from that, the SPECIALIST algorithm is only designed to resolve a jam wave using VSLs. Schelling et al. [2011] integrated the SPECIALIST algorithm with RM. Although it is expected that this may increase the effectiveness, it is also likely that further extending the algorithm to more generic situations is theoretically challenging. This could be addressed by incorporating the VSL control principles used in SPECIALIST in an optimization framework, for instance, using parameterization. This could reduce the computation time while simultaneously improving transparency.

The most relevant scientific insight for the development of *urban* traffic control algorithms in this dissertation is the development of the link transmission model (LTM) by Yperman [2007]. This model is capable of modeling the most relevant traffic dynamics, namely, forward and backward propagating waves, and the saturation flow of queues. In contrast to the commonly used cell transmission model (CTM) proposed by Daganzo [1995], it has the advantage that it does not require to divide a link into segments so that the model is more efficient from a computational point of view. However, not many control algorithms based on the LTM exist. See, Hajiahmadi et al. [2015b] for a model predictive control (MPC) strategy based on the LTM for integrated control of VSLs and RM.

1.3 Research objective

This dissertation addresses the challenge of improving the trade-off between road traffic network performance and required computation time of traffic control algorithms. This is realized by developing algorithms for the control of medium-to-large scale urban and freeway networks. These algorithms are designed in the context of the characteristics, challenges, and opportunities of the network-wide traffic control problem as discussed above.

To this end, the main aim of this dissertation is **the design of computationally efficient traffic control algorithms for throughput improvement of medium-to-large scale freeway or urban traffic networks** that:

- coordinate the control actions of (different types of) actuators at different locations in the network,
- take the impact of the control actions on the network-wide performance over a time horizon into account.

1.4 Research scope

Network-wide traffic control is a challenging problem with many open issues that need to be addressed. The algorithms proposed in this dissertation are meant as a step into the development of the next generation network-wide traffic control algorithms. The main step taken in this dissertation is exploiting the most recent technological innovations and scientific insights in order to realize a better trade-off between computation time and realized performance of network-wide traffic control algorithms. The following scope is considered when developing network-wide traffic control algorithms in this dissertation.

This dissertation focuses on the design of algorithms for **medium-to-large scale traffic networks.** This simplifies the control problems that are to be solved within the sub-networks, since, the number of controlled actuators is reduced. Additionally, a sub-network either consists of urban roads or freeway which simplifies the problem as well. A medium-to-large scale freeway network is defined as a network consisting of tens of kilometers of freeways, tens of VMS gantries, and several RM installations. A medium-to-large scale urban network consists of tens of intersections. This dissertation does not address the problem of coordinating the flows between sub-networks. The reader is referred to [Hajiahmadi et al., 2015b, Zhou et al., 2016] for control approaches aiming at coordinating the flows exchanged between sub-networks.

This dissertation focuses on the design of algorithms for the following **existing traffic control measures:** (in-vehicle) VSLs, RM installations, traffic lights, and route guidance. The reason being that the dissertation mainly aims at optimizing the flows in a

network for which these control measures are specifically suited. Additionally, these measures are also most relevant from a practical point of view.

This dissertation mainly aims at **improving the throughput**. Improving the throughput is one of the most important traffic control objectives. Apart from throughput, safety is typically considered in the design, for instance, by constraining the optimization problem. However, it will not be systematically assessed whether the safety will be improved by applying the algorithms in practice. Other performance indicators, such as, equity or pollution, are not considered in this dissertation.

The algorithms proposed in this dissertation are designed to control **normal traffic** or more specifically, traffic flows consisting of a mix of cars and trucks. The application of the algorithms to networks used by more types of traffic, such as, bikes, pedestrians, public transportation, and emergency vehicles is beyond the scope of this dissertation. It should be noted that including more types of traffic, also called modes, may require different optimization algorithms. For instance, the algorithms may need to be adjusted to maximize the throughput in terms of total travel time of persons instead of vehicles.

This dissertation considers an ideal world in which **no measurement noise** and **no demand prediction uncertainties** are present. In this way, no observers or filters are needed to improve the measurements fed to the controllers so that the simulation results are not biased by measurement errors. Currently, predictions of the demand are obtained using historical data and real-time inductive detector loop measurements. In the near-future, these predictions may be improved using FCD.

1.5 Research approach

The main research objective is achieved by developing several algorithms for the control of traffic in freeway and urban traffic networks. In general, the novelty of these algorithms is in the use of new detection and actuation possibilities or in the use of recent scientific insights to develop more efficient traffic control algorithms. This dissertation is divided into two parts as shown in Figure 1.2. The first part presents algorithms for the control of freeway traffic, and the second part presents algorithms for the control of urban traffic. An overview of the different proposed control algorithms is presented below.

1.5.1 Freeway traffic control

The **first** part of this dissertation aims at developing algorithms for improving the throughput of freeway traffic networks. This part *first* focuses on the use of in-vehicle technologies enabling cooperative systems to improve the freeway throughput. As motivated in Section 1.2.1, using cooperative systems instead of infrastructure based technologies – such as, inductive loop detectors and VSLs displayed on VMSs – can

lead to more efficient traffic control strategies. Most control algorithms proposed in the literature that use in-vehicle technology or cooperative systems focus on the control of individual vehicles or platoons of vehicles using (cooperative) adaptive cruise control ((C)ACC) to stabilize the traffic flow or to allow shorter headways between vehicles. However, much less algorithms for the coordinated control of individual vehicles on an entire freeway stretch have been developed.

Hence, the aim of **Chapter 2** is to *develop a VSL control algorithm that uses individual vehicles as detectors and actuators for coordination of the speed of individual vehicles to improve the freeway throughput.* The insights into the application of shock wave theory to describe the effect of VSLs on the freeway flow will be applied in this chapter. The control of individual vehicles implies that the controller has to compute the control actions for a lot of actuators, namely, all the vehicles on the freeway. Therefore, the controller is designed to require only little computation time. The controller is evaluated using microscopic simulation.

While exploiting in-vehicle technology enabling cooperative systems is one way to improve the performance of freeway traffic control strategies, the application of control strategies that optimize the flows between different network elements – e.g. on-ramps, off-ramps, bottlenecks, and segments – has the potential to improve the freeway performance as well as discussed in Section 1.2. One of the main issues of this type of algorithms is balancing the required computation time and performance of the control strategy. Typically, speeding up the optimization allows to (1) update the control signal more frequently which allows to correct prediction errors more rapidly or (2) to include more complex prediction models that may lead to a better performance.

To this end, the aim of **Chapter 3** is *the development of a computationally efficient model-based predictive control (MPC) strategy for coordinating VSLs and RM installations in order to improve the freeway throughput.* The computational efficiency is improved by reducing the dimension of the optimization problem. This is realized by a spatial discretization of the network into segments, and by exploiting the insights gathered into the application of shock wave theory to describe the effect of VSLs onto the freeway flow to simplify the control problem. The controller performance is evaluated using macroscopic simulation.

1.5.2 Urban traffic control

The **second** part of this dissertation aims at developing algorithms for improving the throughput of urban road networks. This is a complex problem due to the discontinuous nature of the intersection flows, the large number of actuators, and the characteristics of the urban traffic dynamics. To the best knowledge of the author, a computationally efficient optimization algorithm for the coordination of intersection flows that can realize good performance in all traffic regimes is currently lacking.

Therefore, the aim of **Chapter 4** is to *develop an efficient MPC strategy for optimizing the traffic flows that cross the intersections* in order to improve the urban road network throughput. The proposed MPC strategy uses the LTM as the prediction model and aggregates the traffic flow dynamics to tens of seconds so that, instead of green-times, the fractions of green-time used by every stream are the optimization variables, which are continuous. The approach is tested using macroscopic simulation and compared to other, comparable strategies.

The use of in-vehicle technology enabling cooperative systems, or more specifically in-car navigation devices, may cause an interaction effect between the chosen intersection control strategy, and the route choice of the road-users. In order to maximize the network performance, a control strategy has to account for the impact of the control signals onto the route choice and potentially control the route choice itself. However, jointly optimizing the signal timings and route choice is a computational complex problem.

The aim of **Chapter 5** is to *develop a computationally efficient optimization algorithm for the control of intersection flows and route choice* to improve the urban network throughput. This is realized by extending the MPC strategy proposed in Chapter 4. The inclusion of the route choice leads to a non-linear optimization problem so that an efficient optimization algorithm has to be developed. The approach is evaluated using macroscopic simulation.

The algorithms proposed in Chapter 4 and Chapter 5 both assume that the traffic flows at intersections are continuous. However, as explained in Section 1.1, intersection flows are discontinuous by definition. Directly optimizing the signal timings leads to a discontinuous optimization problem which is not real-time feasible when applied to medium-to-large scale networks. In order to apply the control signals computed by the algorithms proposed in Chapter 4 and Chapter 5 they need to be translated to signal timings that are applicable to traffic lights.

Hence, the aim of **Chapter 6** is to *develop a hierarchical control framework to coordinate the signal timings* in order to improve the urban network throughput. The framework consists of two layers. The top layer uses the MPC strategy proposed in Chapter 4 to optimize the aggregated flows at intersections. Next, the bottom layer has to control the signal timings so that the optimized flows are tracked as good as possible. The algorithm is tested using both macroscopic and microscopic simulation.

1.6 Contributions

This dissertations contributes to the scientific literature in several ways. The contributions presented here are summaries of the detailed contributions presented in the introductions of the different chapters.

A theory and algorithm is proposed to resolve a jam wave using FCD and by influencing the speed of individual vehicles on the freeway in Chapter 2. Special attention is paid to satisfy the properties and limitations imposed when implementing cooperative systems, such as, privacy and safety. Additionally, an evaluation is carried out in order to test the performance and behavior of the algorithm.

Insight is gathered into the application of in-vehicle technologies for coordinating the speed of vehicles on an entire freeway to improve the freeway throughput in Chapter 2. The availability of in-vehicle technology enabling cooperative systems is rapidly increasing and most practical applications focus on the application of in-vehicle technology or cooperative systems for the control of an individual vehicle, or in some cases on the control of a platoon of vehicles. Coordinating vehicles on an entire freeway is the next step to which this dissertation contributes.

The balance between computation time and performance of optimization-based network-wide traffic control algorithms is improved in Chapter 3, Chapter 4, and Chapter 5. The algorithms are designed for optimizing the flows in freeway and urban networks by controlling VSLs, RM installations, flows at intersections, and route guidance to improve the network-wide throughput. Optimization-based algorithms can help to make more efficient use of the network capacity. The work in these chapters provide a step in the application of optimization-based control algorithms by reducing the computation time of these algorithms.

An efficient approach for optimizing the VSL values and RM rates on a stretch of freeway over a time horizon is proposed in Chapter 3. The algorithm is efficient due to the novel parameterization of integrated VSL and RM control strategies. Macroscopic simulations show the improved efficiency due to the parameterization.

A linear optimization approach based on the LTM is proposed for optimizing the aggregated intersection flows in Chapter 4. The optimization approach is designed for a MPC strategy. Compared to existing linear optimization approaches, the approach is capable of accounting for upstream and downstream propagating shock-waves and saturated traffic flows while having a better balance between computation time and realized performance. Macroscopic simulations show the improved balance between computation time and performance when compared to other comparable strategies.

An efficient algorithm for optimizing the aggregated traffic flows and routing decisions in an urban traffic network is presented in Chapter 5. A major element of the optimization algorithm is the analytic approximation of the gradient that is proposed in this chapter. The performance of the algorithm is tested using macroscopic simulation.

A real-time feasible, hierarchical control framework for the control of signal timings is proposed in Chapter 6. The framework is designed to improve the network-wide urban network throughput in all traffic regimes. The proposed framework is evaluated using both macroscopic and microscopic simulation.

An efficient framework to control the signal timings, and coordinate the flows at different intersections in an urban network is presented in Chapter 6. The framework

contributes to the application of optimization-based network-wide traffic control algorithms by reducing the required computation time. The algorithm is tested using the MPC strategy presented in Chapter 4 but can be extended to include other optimization-based algorithms as well.

1.7 Dissertation outline

Figure 1.2 presents the dissertation outline and the relations between the chapters. This dissertation is divided into two parts. The **first** part presents algorithms for the control of freeway traffic. Chapter 2 first presents a cooperative speed control algorithm to resolve jam waves on the freeway. Next, Chapter 3 presents an efficient optimization algorithm for the coordination of flows exchanged between different elements of a freeway network. The **second** part of this dissertation presents algorithms for the control of urban traffic. Chapter 4 presents a linear optimization procedure to optimize the flows in an urban network. Chapter 5 extends the approach proposed in Chapter 4 by including the control of routing decisions in the optimization problem. Chapter 6 presents a hierarchical control framework for the coordination of signal timings which uses the approach proposed in Chapter 4 in a top layer to optimize the flows in the network while the bottom layer is used to translate the optimized flows into signal timings. Chapter 7 concludes this dissertation.

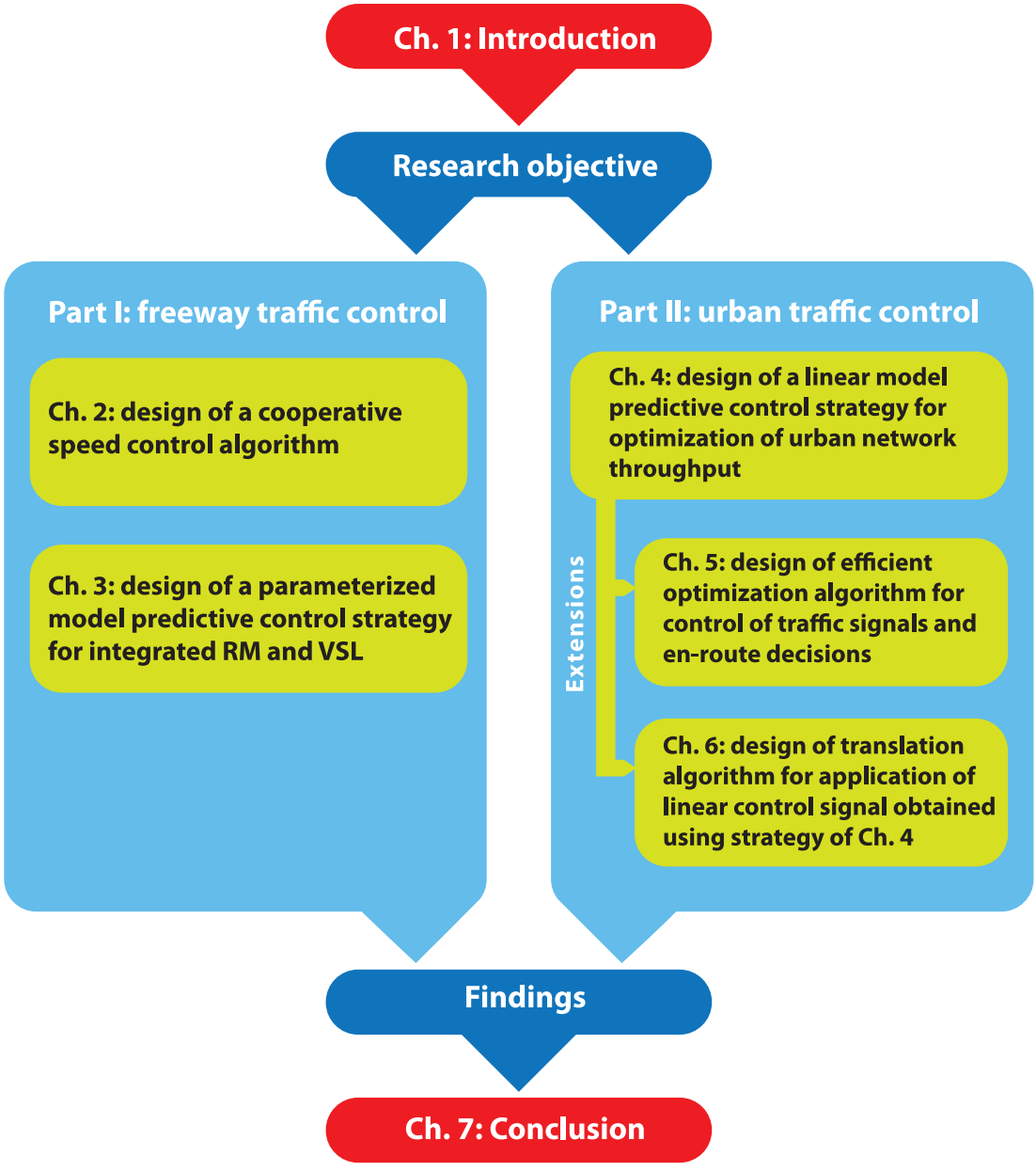


Figure 1.2: Overview of the dissertation

Part I

Freeway traffic control

Chapter 2

COSCAL v1: A cooperative speed control algorithm for resolving jam waves

---

In this chapter an approach is developed to use in-vehicle technology to improve the freeway throughput by coordinating the speed of individual vehicles on a freeway stretch. This chapter is based on the following paper that is currently being prepared for submission:

G.S. van de Weg, A. Hegyi, S.E. Shladover, X.-Y. Yun, D. Chen, and S.P. Hoogendoorn, COSCAL v1: A cooperative speed control algorithm for resolving jam waves. To be submitted.

Abstract

In this paper, an algorithm for cooperative systems is developed and evaluated which improves the freeway throughput by resolving a jam wave, i.e., a jam that travels in the opposite direction of traffic. This algorithm – called COSCAL v1 – determines speed instructions for individual vehicles based on speed and position data of individual vehicles.

The speed instructions are formulated as driving tasks, or modes, which relate to the task a vehicle has to perform in order to resolve a jam wave, such as, autonomous driving, slowing down for jam resolution, or slowing down for stabilization. These tasks are communicated in such a way that a low communication bandwidth is required. Besides that, the communication is formulated in such a way that the privacy of the users is respected.

The algorithm has been tested using the micro-simulation package Vissim. The evaluations showed that the algorithm can resolve a jam wave on a single lane freeway resulting in a TTS gain of 7.3% and that the algorithm is also capable of resolving a jam wave on a two lane freeway resulting in an average TTS gain of 17.3%. It is shown that the behavior of the algorithm is similar to the behavior of SPECIALIST. Finally, it is discussed how this algorithm can be extended to deal with lower penetration rates, a combination of in-vehicle and road-side technologies, and multiple on-ramps and off-ramps.

2.1 Introduction

The current proliferation of in-vehicle technologies – e.g. on-board computers or GPS navigation devices – introduces opportunities for better dynamic traffic management (DTM) of freeway traffic when compared to the currently used infrastructure-based systems – i.e., systems using road-side detection and actuation, such as inductive detector loops and variable message sign gantries. The reason for this is that DTM based on in-vehicle technologies has several advantages, such as: higher resolution traffic data, higher control freedom, and reduced dependency on costly infrastructure-based systems. Therefore, this paper focuses on the use of in-vehicle technology for DTM to improve the freeway performance. More specifically, this paper focuses on cooperative systems which are systems in which vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communication is enabled.

A well-known DTM measure to improve the freeway performance is the use of variable speed limits (VSLs). Research has shown that VSLs can be used to improve, among other things, the freeway safety and throughput. One way of improving the freeway throughput using VSLs is by reducing the impact of the capacity drop caused by congestion. The capacity drop is a term that refers to the phenomenon that downstream of congestion the flow is lower than the free-flow capacity of the freeway. The capacity drop is also observed with jam waves i.e., a form of congestion of which the head propagates upstream – and can be up to 30% [Kerner and Rehborn, 1996].

Hence, the aim of this paper is the development and evaluation of a cooperative VSL control strategy that improves the freeway throughput by reducing the impact of the capacity drop. Several conditions have to be satisfied when applying such a strategy. These conditions can be divided into conditions for the application of VSLs and conditions for the application of cooperative systems as discussed below. These conditions are used when studying the literature in the next subsection and when designing the control strategy. It must be noted that satisfying all these conditions is rather challenging. Therefore, Section 2.2.1 presents the design considerations that are accounted for in this paper.

The following conditions have to be satisfied when applying VSLs in practice. First of all, authorities typically only allow the application of a single or a small set of VSL

values. Secondly, the VSLs that are imposed to the road users should not lead to unsafe situations. One example of an unsafe situation is a situation in which only part of the traffic receives a reduced speed limit advice. This could lead to large speed differences between uninformed and informed road users which increases the possibility of unsafe situations. Thirdly, the system should be comfortable for the user. An example of an uncomfortable situation is when a road user is experiencing rapidly fluctuating VSL advice. Finally, the road users may not fully comply to the displayed VSLs so that the algorithm has to account for possible non-compliance.

Applying in-vehicle technologies for DTM is subject to several conditions as well. First of all, in-vehicle technology enables the use of floating car data (FCD) for DTM. This data contains privacy sensitive information, such as, the location of the users over time. For privacy reasons it is not feasible to track the location of individual vehicles over time. Secondly, it is expected that in the coming years only low percentages of vehicles equipped with in-vehicle technology can be used for DTM. This could negatively affect the effectiveness of such a system. Thirdly, cooperative systems may consists of several hundreds or thousands of vehicles. This could potentially require a lot of communication bandwidth which would make the system expensive or degrade the performance of the communication system. Therefore, a cooperative systems based DTM algorithm should only require low communication bandwidth. Finally, various types of in-vehicle systems are expected to co-exist, e.g. adaptive cruise control (ACC), cooperative ACC (CACC), or in-vehicle messages. Thus, the system has to be able to deal with various types of actuation possibilities.

2.1.1 Literature review

Two main approaches for improving the freeway throughput by means of infrastructure based variable speed limits can be identified [Hegyi et al., 2009]. The first is homogenization which means that a speed limit is shown in order to reduce the speed of some of the vehicles such that speed differences between vehicles are reduced [Smulders, 1990, Kühne, 1991, Van den Hoogen and Smulders, 1994]. The idea is that this removes disturbances which may cause congestion. Hence, by homogenizing the speeds it is expected that the throughput improves [Smulders, 1990]. However, this effect was not observed during field-tests [Van den Hoogen and Smulders, 1994].

The second approach uses speed limits to reduce the flow on the freeway. Several algorithms exist that exploit this effect. Carlson et al. [2011] use variable speed limits to gate traffic that is entering a bottleneck in their approach called mainstream traffic flow control (MTFC). The authors impose a variable speed limit at a fixed location upstream of a bottleneck and adjust the speed limit in such a way that congestion upstream of the bottleneck is created. By adjusting the value of the VSL the authors can control the outflow out of the controlled congestion in such a way that it is near the capacity of the bottleneck. In this way, congestion at the bottleneck can be prevented

or postponed such that the impact of the capacity drop in the bottleneck is reduced. The approach was tested using simulation studies.

Hegyi et al. [2010] proposed an algorithm called SPECIALIST in which VSLs are used to resolve a jam wave – i.e., congestion with a length of roughly 1 to 2 km that propagates in the upstream direction of the freeway. The SPECIALIST algorithm detects a jam wave using inductive detector loops as indicated with task I in Figure 2.1. When it assesses this jam wave as resolvable it first applies a pre-defined VSL value instantaneously over a freeway stretch directly upstream of the jam wave as indicated with the line between points B and C in Figure 2.1. This is called task II; jam resolution. Next, VSLs are imposed upstream of the speed-limited area along the line between points C and E in Figure 2.1 to stabilize the traffic flow - by creating a stable combination of speed and density - that is approaching the speed-limited area. This is called task III; stabilization. This causes a reduction of the flow into the jam wave so that it can resolve without triggering an upstream congestion. After the jam wave is resolved, the traffic in the speed-limited area can be released and a higher freeway flow can be achieved since the capacity drop is no longer present as indicated with the line between poitns D and E in Figure 2.1. It follows from shock wave theory that the density and flow in (and downstream of) the speed-limited area can be controlled by adjusting the speed with which the upstream (and downstream) boundary of the speed-limited area propagates [Lighthill and Whitham, 1955]. SPECIALIST was tested on the A12 freeway in the Netherlands and it was found that it is capable of resolving jam waves and stabilizing traffic, resulting in improved freeway throughput. A challenge of the SPECIALIST algorithm is that it has a feed-forward structure so that it cannot adjust its control action to unanticipated changes in the traffic situation.

Chen et al. [2014] propose an approach to resolve congestion at a bottleneck. In their approach VSLs are initially imposed upstream of the bottleneck in the congested area in order to move the head of the queue away from the bottleneck. Then, the bottleneck outflow can be increased, since, the capacity drop is no longer present. After that, the value of the VSL is increased in order to match the outflow out of the speed limited area to the bottleneck capacity. To the best knowledge of the authors, the approach was not evaluated using simulations.

Several researchers have studied the extension of infrastructure based DTM with in-vehicle technology. Heygyi et al. [2013] investigated the use of in-vehicle systems to enhance the infrastructure based SPECIALIST algorithm. It was found that even small percentages of equipped vehicles can improve the performance of the SPECIALIST algorithm. The reason for this was that the speed with which jam waves were detected was increased. Grumert et al. [2013] integrated a roadside VSL system with in-vehicle speed limits. The authors found positive effects on the acceleration and deceleration and lower emissions. The main reason for this effect was that vehicles received speed limit advice faster.

Influencing the speed of vehicles using only in-vehicle systems has drawn a lot of attention in recent years. Currently, a popular research topic is the application of CACC

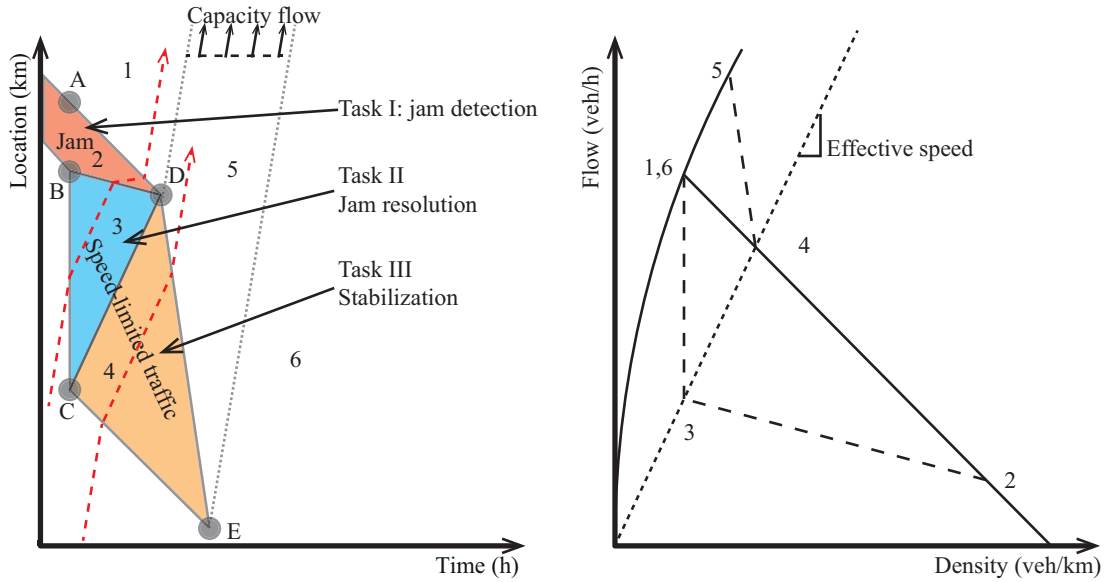


Figure 2.1: Left: Time-space plot with a schematic representation of the tasks needed to resolve a jam wave. Right: Corresponding fundamental diagram. The red dashed line indicates a trajectory of a vehicle that is speed limited for stabilization. The solid lines indicate shock waves between the states that are indicated with numbers. The slope of the shock waves can be derived from the fundamental diagram on the right. The point A is the head of the jam when the algorithm is started. Initially, speed-limits are imposed from point A to point C resulting in a flow drop as can be observed in the fundamental diagram. The line between point C and D is the boundary between the jam resolution and the stabilization area. After the control has started speed limits are gradually extended upstream along the line C–E and when the jam has resolved the speed-limits are gradually released along the line D–E. The flow out of the jam wave (state 1) is lower than the flow out of the speed-limited area (state 5).

that enables communication of acceleration and speed information of multiple vehicles driving close together. The advantage of CACC is that it enables a reduction of the following distance between vehicles. Several studies have shown that high penetration rates of CACC enabled vehicles can lead to increased freeway capacity when compared to manually driven systems or adaptive cruise control systems [Shladover, 2009, Van Arem et al., 2006, Arnaout and Arnaout, 2014, van der Werf et al., 2002]. Despite these positive effects it must be emphasized that in the coming years the penetration rates of CACC vehicles will probably be low. Implying that these effects on capacity will be limited and the focus should be on the transition and co-existing of infrastructure based and in-vehicle systems.

Another challenge of CACC systems is that they mainly focus on the microscopic level, i.e., they focus at the control of a few vehicles or a platoon of vehicles. However, when controlling traffic using in-vehicle technologies, also the impact on mechanisms in the traffic flow on a macroscopic level should be considered. Wang et al. [2015] integrated the SPECIALIST control algorithm to give driving instructions to ACC equipped vehicles in order to resolve a jam wave. The reason why this was required is that different driving strategies are required in time and space when resolving a moving jam, hence the need for a coordinating level. Another example is the work of Scarinci et al. [2013] who reduced the speed of cooperative systems enabled vehicles on the freeway to create gaps on the freeway for traffic merging from a metered on-ramp. Another noteworthy example is the work of Nishi et al. [2013] who showed that a single vehicle can resolve a jam wave. However, in their approach effects of safety and stabilization, as used in the SPECIALIST algorithm are not included.

Concluding, a lot of research has investigated the use of VSLs to improve freeway throughput by reducing the impact of the capacity drop. Several studies have shown the potential benefits of using in-vehicle systems to enhance the performance of infrastructure based DTM measures. Also, there is a need for coordinating algorithms when developing DTM measures based on cooperative systems.

2.1.2 Contribution and approach

In this paper a cooperative speed control algorithm is proposed that improves freeway throughput by resolving jam waves. The algorithm uses similar concepts as the SPECIALIST algorithm but the formulation is fundamentally different such that certain limitations of the SPECIALIST algorithm are solved. The main contributions of this paper are that 1) a theory is proposed to resolve a jam wave and stabilize traffic using a cooperative system in Section 2.3, 2) an algorithm is proposed to apply the theory while satisfying the constraints imposed by VSL and cooperative systems in Section 2.4, and 3) insight into the behavior and performance of the algorithm is obtained via microscopic simulation Section 2.5. Compared to previous works on cooperative systems as discussed in Section 2.1.1, an advantage of this approach is that it allows the coordination of the macroscopic effects of cooperative systems. Compared to the previous

works on SPECIALIST, an advantage is that the proposed algorithm has a feedback structure, uses FCD for detection, and the individual vehicles for actuation.

The algorithm proposed in this paper is called COSCAL v1 which is an acronym for 'cooperative speed control algorithm version 1'. The algorithm is designed for a system in which every vehicle is equipped with in-vehicle technology which is further motivated in Section 2.2.1. An extension of the algorithm to deal with low penetration rates of cooperative vehicles and infrastructure based systems is described by Mahajan et al. [2015] and called COSCAL v2. An integration of COSCAL v1 with ramp metering is detailed in [van de Weg et al., 2014a].

2.2 Overview of the COSCAL v1 strategy

This section provides an overview of the COSCAL v1 strategy. The strategy uses similar concepts as the SPECIALIST algorithm to resolve a jam wave. However, as detailed in Section 2.3, the formulation of the approach is fundamentally different, overcoming several limitations of the SPECIALIST algorithm and allowing the use of cooperative systems. Before giving this overview, the next subsection first introduces the design considerations. Section 2.4 details the implementation of the COSCAL v1 theory within an algorithm.

2.2.1 Design considerations

This section discusses the assumptions and design choices. First the assumptions related to the use of a VSL system and next the assumptions related to the use of cooperative systems are discussed.

The following design choices and assumptions with respect to VSLs are made:

- The algorithm uses a single value of the VSLs to slow down vehicles. This design choice is motivated by the application of the algorithm to mixed traffic situations which requires that the VSL approach is similar to currently used systems. In practice only a limited set of VSL values can be imposed, also in order to resolve a jam wave as quickly as possible the speed limits need to be reduced to the lowest admissible speed, i.e., the lowest speed allowed by the road authorities that does not result in congestion or unsafe situations. Another reason for working with a single value for the VSLs is that it can lead to more driving comfort;

- The algorithm contributes to a safe and comfortable driving by homogenizing the density of traffic that is speed-limited for stabilization. The density should be chosen such that a stable traffic state is created;

- It is assumed that speed-limited vehicles drive, on average, with the effective speed v^{eff} (km/h). This effective speed includes possible non-compliance to the VSLs;

- This paper does not consider possible extensions of the algorithm to other DTM measures, such as, ramp metering. This extension is discussed in Section 2.6 and an approach is proposed in [van de Weg et al., 2014a].

The following design choices and assumptions with respect to cooperative systems are made:

- The algorithm uses V2I communication. There are several reasons motivating this design choice. First of all, it becomes easier to impose speed-limits to multiple vehicles simultaneously. Secondly, in a V2V system a communication delay is introduced, since, every vehicle needs to communicate with all the other vehicles. This delay depends on the number of vehicles that need to communicate and the length over which this communication is required which depends in its turn on the traffic situation. Thirdly, it is easier to integrate a V2I system with other roadside systems;

- For privacy reasons and in order to keep the system simple, the central server cannot track individual vehicles. Therefore, vehicles will store their own trajectory data and act as the 'memory' of the system. Vehicles only communicate their current position and speed information to the central server;

- The central server does not address individual vehicles. Instead, the central server will communicate generalized messages indicating the driving strategy that should be followed on every freeway segment. An example of such a message could be: 'vehicles between position 1 km and 3 km, reduce speed for jam wave resolution';

- Several systems for influencing the speed of vehicles exist, ranging from CACC systems to in-vehicle messages that should be manually implemented by the driver. In order to be compatible with all systems, only speed instructions are given. Other instructions, for instance, keeping a desired headway time are very difficult to be followed up by humans and will, therefore, not be considered in this paper;

- A 100% penetration rate of equipped vehicles is assumed. The reason for this is that the aim of this paper is to design and test the theory for a cooperative algorithm.

The following general design choices and assumptions have been made:

- It is assumed that there is at most one jam;

- The algorithm is designed for a single lane freeway. In Section 2.5.2 the algorithm is also tested when applied to a two-lane freeway and Section 2.6 discusses further research directions for the application of the algorithm to multi-lane freeways.

2.2.2 COSCAL v1 overview

The goal of the COSCAL v1 strategy is to determine the desired driving strategies on different freeway segments so that a jam wave can resolve. This is communicated using generalized messages containing so called actuation lines that define which driving strategy vehicles have to follow on which (time varying) freeway segments as is detailed in Section 2.4. Based on these actuation lines, vehicles can compare their current location with the segments and adopt their speed to the desired driving strategy of the segment in which they are driving.

COSCAL v1 operates both in the vehicles and on the roadside. The algorithm inside the vehicle keeps track based on the FCD of the vehicle whether it is inside of a jam. The detection mode of a vehicle indicates whether it is inside of a jam or not. The in-vehicle algorithm is also used to modify the speed of the vehicles based on the actuation lines. The roadside algorithm computes, based on the speeds, positions, and detection modes of all the vehicles, the current actuation lines.

The COSCAL v1 theory consist of four steps to compute the actuation lines. These steps correspond to the different tasks of SPECIALIST as shown in Figure 2.1. These steps are: (1) jam wave detection, (2) initial speed limitation for jam resolution, (3) speed limitation for stabilization, and (4) speed limit release. The jam wave detection is done by the individual vehicles which keep track of their speed as discussed in Section 2.3.1. Next, the algorithm finds the most upstream vehicle that has to be speed-limited to resolve the jam as detailed in Section 2.3.2. The algorithm then decides which vehicles, upstream of this former vehicle have to be speed-limited for stabilization according to the theory detailed in Section 2.3.3. This is done in such a way that a constant following distance between vehicles is realized on the average so that a stable density is realized as shown in Section 2.3.5. When the jam has been resolved, the algorithm determines which vehicles should be released from the speed-limited area as described in Section 2.3.4.

2.3 COSCAL v1 theory

Now that the main approach has been detailed and all the design choices and assumptions have been introduced the theory can be described. In the description below, vehicle index i is used to refer to individual vehicles, where the more downstream vehicle has a lower index. The discrete time index k_i, refers to the time period $[k_i T_i, (k_i+1)T_i)$,

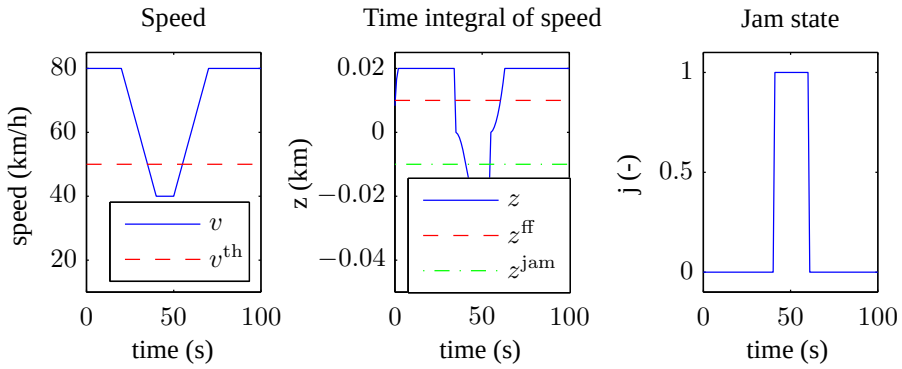


Figure 2.2: Example of the jam tracking approach for $vth = 50$ km/h, $z^{\text{ff}} = 10/1000$ km, and $z^{\text{jam}} = -10/1000$ km.

where T_i (h) is the discrete time step size of the system in the vehicle. Time step T_i typically has a value of less than a second. In addition, a roadside time step T^{rs} with index k^{rs} refers to the roadside control system.

2.3.1 Step I: Jam detection

In COSCAL v1 the jam is detected by individual vehicles. A jam is always associated with low speed, and thus a jam is detected if the vehicle speed $v_i(k_i)$ (km/h) is below a certain threshold v^{th} (km/h) for a sufficiently long time. Similarly the detected jam state is restored to free flow if the speed is above the threshold for a sufficiently long time. In order to determine what is sufficiently long, the time integral $z_i(k_i)$ (km/h) is taken of the difference $v_i(k_i) - v^{\text{th}}$ as long as the speed $v_i(k_i)$ remains continuously above or below v^{th}, and the integral is compared with thresholds z^{ff} (km) (free flow) and z^{jam} (km) (jam), with $z^{\text{jam}} < 0 < z^{\text{ff}}$ according to

$$\tilde{z}_i(k_i) = \begin{cases} z_i(k_i - 1) + T_i(v_i(k_i) - v^{\text{th}}) & \text{if } (z_i(k_i - 1) \geq 0 \wedge v_i(k_i) > v^{\text{th}}) \vee \ldots \\ & \quad (z_i(k_i - 1) \leq 0 \wedge v_i(k_i) < v^{\text{th}}) \\ T_i(v_i(k_i) - v^{\text{th}}) & \text{otherwise} \end{cases} \tag{2.1}$$

$$z_i(k_i) = \min\left(2z^{\text{ff}}, \max\left(2z^{\text{jam}}, \tilde{z}_i(k_i)\right)\right) \tag{2.2}$$

$$j_i(k_i) = \begin{cases} 1 & \text{if } z_i(k_i) \leq z^{\text{jam}} \\ 0 & \text{if } z_i(k_i) \geq z^{\text{ff}} \\ j_i(k_i - 1) & \text{otherwise} \end{cases} \tag{2.3}$$

where $\tilde{z}_i(k_i)$ is truncated in (2.2) to prevent that $z_i(k_i)$ grows to plus or minus infinity (to prevent implementation problems), and $j_i(k_i)$ indicates the jam state of vehicle i, where 1 means jam and 0 means free flow. Using the integration and thresholding prevents the chattering of the jam states if the speed closely fluctuates around v^{th}. The approach is illustrated in Figure 2.2.

2.3.2 Step II: Initial speed limitation for jam resolution

After the jam has been detected, the vehicles directly upstream of the jam have to reduce their speed to v^{eff}. In this step, it is determined which vehicle is the most upstream vehicle that needs to be slowed down in order to resolve the jam. To determine this last vehicle, the following reasoning is used.

Let us consider a vehicle that is slowed down and that will join the queue at some point, before the jam is resolved. An example of a trajectory of such a vehicle is shown by the blue dashed line in Figure 2.3. The time t_i^{exit} (h) when vehicle i leaves the jam, can be calculated based on the following assumptions:

- The jam head has a known, constant propagation speed v^{head} (km/h). For jam waves, this is about -18 km/h, for jams at on-ramps this is zero.
- The flow $q^{[1]}$ (veh/h) and the density $\rho^{[1]}$ (veh/km) downstream of the jam over all lanes after the traffic has reached its free-flow speed, are also known and constant. (The superscript [1] refers to the corresponding traffic state in SPECIALIST, see Figure 2.1.) For jam waves this flow equals the queue discharge rate, and is around 70% of the normal free-flow capacity [Kerner and Rehborn, 1996].
- The speed $v^{[2]}$ and density $\rho^{[2]}$ in the jam are also constant. These are typically the jam speed (close to zero) and the jam density (about 100 veh/km).

These assumptions are not very limiting, since there are many empirical observations that support them. Note that the last two entail the first assumption.

Based on these assumptions, the flow that crosses the head of the jam can be calculated using the same reasoning as Lighthill and Whitham [1955] used for the derivation of front speeds. A moving observer who moves together with the head of the jam will see not only a flow $q^{[1]}$ (or $q^{[2]}$), depending on on which side of the front the observer is looking), but also the vehicles that the observer is passing, due to his own speed. So the total flow q^{head} (veh/h) the observer sees is given by

$$q^{\text{head}} = q^{[1]} - v^{\text{head}}\rho^{[1]}\,, \tag{2.4}$$

or equivalently by

$$q^{\text{head}} = q^{[2]} - v^{\text{head}}\rho^{[2]}\,. \tag{2.5}$$

Now, the time t_i^{exit} (h) when vehicle i will exit the queue can be calculated using the number of vehicles $N_i(k^{\text{rs}})$ (veh) between the first (most downstream) vehicle in the queue and vehicle i. The exit time is given by

$$t_i^{\text{exit}}(k^{\text{rs}}) = \frac{N_i(k^{\text{rs}})}{q^{\text{head}}}\,. \tag{2.6}$$

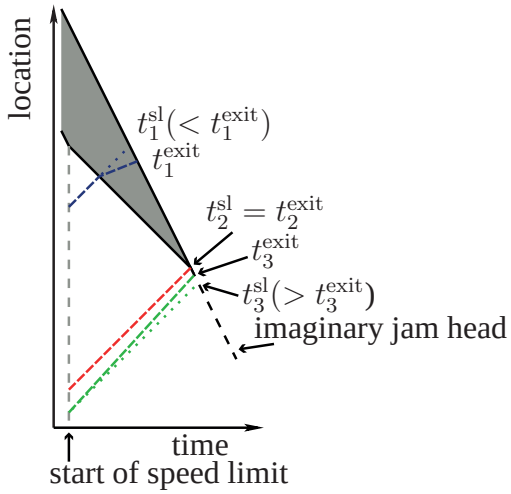


Figure 2.3: Different vehicle trajectories and the corresponding t^{exit} and t^{sl}. The blue vehicle (blue dashed line) has to join the queue even if it is slowed down, and therefore its average speed will be lower than the effective speed limit. The blue dotted line is the trajectory if the vehicle would be able to maintain a speed equal to v^{eff}. The red vehicle (both dashed and dotted) arrives at the head of the queue exactly while it maintains a speed equal to v^{eff} without having to slow down (i.e., without joining the queue). The green vehicle avoids the queue if it maintains v^{eff} (dotted). In fact it could even travel somewhat faster to cross the line of the imaginary jam head at the time based on (2.6). The last vehicle that should be slowed down is the red one.

This time implicitly includes a partial free-flow travel (up to the tail of the queue) and a queuing part, as indicated by the dashed blue line in Figure 2.3. This equation holds as long as vehicle i joins the queue before it exits the queue. Note that (2.6) implies that the exact trajectory of the vehicle is irrelevant as long as the vehicle will join the queue at least for a moment.

Now, if vehicle i slows down due to speed limitation, and slows down sufficiently then it may not have to join the queue. Not joining the queue is equivalent to saying that the queue has resolved downstream of vehicle i. This is the case when the vehicle crosses the imaginary head of the queue later than it would do so based on (2.6). This is indicated with the green dashed line that crosses the black dotted line in Figure 2.3.

If it is assumed that vehicle i will travel at speed v^{eff} after it has been slowed down, then the time $t_i^{\mathrm{sl}}(k^{\mathrm{rs}})$ when its trajectory will cross the (imaginary) trajectory of the queue front is given by:

$$t_i^{\mathrm{sl}}(k^{\mathrm{rs}}) = \frac{x^{\mathrm{head}}(k^{\mathrm{rs}}) - x_i(k^{\mathrm{rs}})}{-v^{\mathrm{head}} + v^{\mathrm{eff}}}\,, \tag{2.7}$$

where $x^{\mathrm{head}}(k^{\mathrm{rs}})$ is the location of the jam head at time step k^{rs}. The corresponding trajectory for the blue vehicle is indicated by the blue dotted line.

For vehicles that will join the queue, it holds that

$$t_i^{\mathrm{exit}}(k^{\mathrm{rs}}) > t_i^{\mathrm{sl}}(k^{\mathrm{rs}})\,, \tag{2.8}$$

such as the blue vehicle in Figure 2.3. For the vehicle that joins the queue at the moment that the queue is being resolved (red vehicle in Figure 2.3), it holds that

$$t_i^{\mathrm{exit}}(k^{\mathrm{rs}}) = t_i^{\mathrm{sl}}(k^{\mathrm{rs}})\,, \tag{2.9}$$

and for the vehicles that will not join the queue anymore (green vehicle), it holds that

$$t_i^{\mathrm{exit}}(k^{\mathrm{rs}}) < t_i^{\mathrm{sl}}(k^{\mathrm{rs}})\,. \tag{2.10}$$

Using this, each individual vehicle can be checked starting from the first vehicle directly upstream of the jam for the following condition:

$$t_i^{\mathrm{exit}}(k^{\mathrm{rs}}) \leq t_i^{\mathrm{sl}}(k^{\mathrm{rs}})\,. \tag{2.11}$$

Then the most upstream vehicle j that should be slowed down, is the first vehicle for which (2.11) holds. In Figure 2.3 this is the red vehicle.

The current position $x_j(k^{\mathrm{rs}}T^{\mathrm{rs}})$ of the jam resolving vehicle and the effective speed v^{eff} defines the R-tail line $L^{\mathrm{R-tail}}(k^{\mathrm{rs}})$:

$$L^{\mathrm{R-tail}}(k^{\mathrm{rs}}) = \{x_j(k^{\mathrm{rs}}T^{\mathrm{rs}}), k^{\mathrm{rs}}T^{\mathrm{rs}}, v^{\mathrm{eff}}\}\,, \tag{2.12}$$

which is used to communicate the upstream end of the area R in which vehicles are speed-limited for jam resolution as detailed in Section 2.4. Additionally, if the R-tail line exists, an R-head line $L^{\mathrm{R-head}}(k^{\mathrm{rs}})$ is defined by the position $x_i(k^{\mathrm{rs}}T^{\mathrm{rs}})$ (km) of the most downstream vehicle in the jam, and the speed v^{head} of the head of the jam:

$$L^{\mathrm{R-head}}(k^{\mathrm{rs}}) = \{x_i(k^{\mathrm{rs}}T^{\mathrm{rs}}), k^{\mathrm{rs}}T^{\mathrm{rs}}, v^{\mathrm{head}}\}\,. \tag{2.13}$$

2.3.3 Step III: Speed limitation for stabilization

The vehicles following the most upstream vehicle that is slowed down to resolve the jam, or the most downstream vehicle in the stabilization area when the jam has resolved, should realize the target density $\rho^{[4]}$ (veh/km) in addition to the target speed v^{eff}. In microscopic terms this means that the following distance d^{headway} should be

$$d^{\mathrm{headway}} = 1/\rho^{[4]}\,, \tag{2.14}$$

on the average. The density is a tuning variable and is chosen such that it corresponds to stable traffic.

This density is realized by properly slowing down the vehicles that are in free flow upstream of already speed-limited vehicles. At each time $t^{\mathrm{rs}} = k^{\mathrm{rs}}T^{\mathrm{rs}}$ a reference vehicle j^{ref} is determined upstream of which the density $\rho^{[4]}$ should be realized. In the case that there is a jam, this vehicle is the vehicle j that should be speed-limited to resolve the jam according to (2.11). In the case that the jam has resolved, it is the first

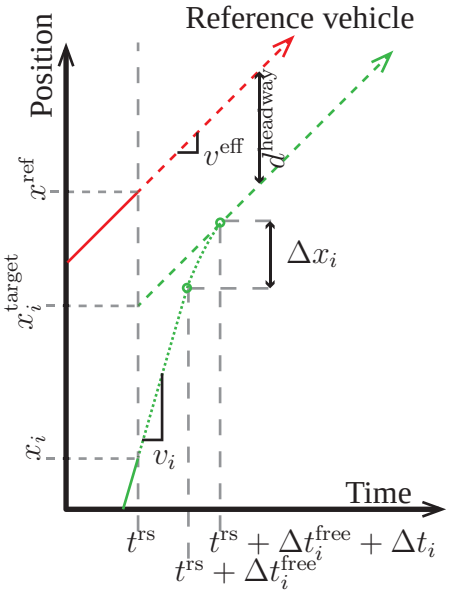


Figure 2.4: Vehicle i will reach the target trajectory which lies a distance of d^{headway} km upstream of the reference trajectory x^{ref} by starting to decelerate at time $t^{\text{rs}} + \Delta t_i^{\text{free}}$.

vehicle that is upstream of the S-head line – i.e., the line describing the downstream end of the stabilization area – that will be defined in Section 2.3.4.

Note that the reference vehicle j^{ref} is at location $x_j^{\text{ref}}(t^{\text{rs}})$ (km) and is traveling with speed $v_j^{\text{ref}}(t^{\text{rs}})$ (km/h). The vehicle is speed-limited and in the case that it has not reached the effective speed v^{eff} yet it will do so after time $\Delta t_j^{\text{ref}}(t^{\text{rs}})$ (h) and distance $\Delta x_j^{\text{ref}}(t^{\text{rs}})$ (km) given as:

$$\Delta t_j^{\text{ref}}(t^{\text{rs}}) = \frac{\max\{v^{\text{eff}}, v_j^{\text{ref}}(t^{\text{rs}})\} - v^{\text{eff}}}{a^{\text{T}}} \tag{2.15}$$

$$\Delta x_j^{\text{ref}}(t^{\text{rs}}) = \frac{1}{2}(\max\{v^{\text{eff}}, v_j^{\text{ref}}(t^{\text{rs}})\} + v^{\text{eff}})\Delta t_j^{\text{ref}}(t^{\text{rs}})\,. \tag{2.16}$$

The reference trajectory that this vehicle determines is then defined by the speed v^{eff} and position $x^{\text{ref}}(t^{\text{rs}})$ (km) given as:

$$x^{\text{ref}}(t^{\text{rs}}) = x_j^{\text{ref}}(t^{\text{rs}}) + \Delta x_j^{\text{ref}}(t^{\text{rs}}) - v^{\text{eff}}\Delta t_j^{\text{ref}}(t^{\text{rs}})\,, \tag{2.17}$$

that together define the following reference line:

$$x^{\text{ref}}(t) = x^{\text{ref}}(t^{\text{rs}}) + (t - t^{\text{rs}})v^{\text{eff}}\,. \tag{2.18}$$

The idea is that every vehicle i upstream of vehicle j^{ref} should reach its own target trajectory line defined by the position $x_i^{\text{target}}(t^{\text{rs}})$ given as:

$$x_i^{\text{target}}(t^{\text{rs}}) = x^{\text{ref}}(t^{\text{rs}}) - N_{j^{\text{ref}}-i}d^{\text{headway}}\,, \tag{2.19}$$

where $N_{j^{\text{ref}}-i}$ (veh) is the number of vehicles between vehicle j^{ref} and vehicle i as is illustrated in Figure 2.4.

This is realized by checking whether a vehicle will need to start decelerating during the current time step. This is done iteratively in the upstream direction starting from the vehicle upstream of the vehicle i^{us} (-). Vehicle i^{us} is the last vehicle in the 'platoon' directly upstream of j^{ref} that is traveling with a speed $v_{i^{\text{us}}}(t^{\text{rs}}) \leq v^{\text{eff}} + \epsilon^{v^{\text{eff}}}$, where $\epsilon^{v^{\text{eff}}}$ (km/h) is a threshold. If no such vehicle exists, then it is equal to the reference vehicle j^{ref}.

It might be the case that the vehicle i^{us} has decelerated too fast and is traveling too far – i.e., more than a threshold $\gamma d^{\text{headway}}$ (km) with γ (-) a tuning parameter representing a fraction of the following distance – upstream of its target trajectory line. This may prevent the next upstream vehicle from reaching its target trajectory line (simply because the vehicle is blocked), and so on. In the worst case, this process may continue for several vehicles, leading to a local accumulation of vehicles, and to a possible breakdown. Therefore, the reference line $x^{\text{ref}}(t^{\text{rs}})$ and index j^{ref} are reset to the vehicle i^{us} if this happens:

$$x^{\text{ref}}(t^{\text{rs}}) = \begin{cases} x_{i^{\text{us}}}(t^{\text{rs}}) & \text{if } x_{i^{\text{us}}}(t^{\text{rs}}) \leq x_{i^{\text{us}}}^{\text{target}}(t^{\text{rs}}) - \gamma d^{\text{headway}} \\ x^{\text{ref}}(t^{\text{rs}}) & \text{otherwise,} \end{cases} \tag{2.20}$$

$$j^{\text{ref}} = \begin{cases} i^{\text{us}} & \text{if } x_{i^{\text{us}}}(t^{\text{rs}}) \leq x_{i^{\text{us}}}^{\text{target}}(t^{\text{rs}}) - \gamma d^{\text{headway}} \\ j^{\text{ref}} & \text{otherwise.} \end{cases} \tag{2.21}$$

Now, for the vehicles upstream of $i^{\text{us}}$the time $\Delta t_i^{\text{free}}(t^{\text{rs}})$ (h) after which they will need to start decelerating can be calculated. Note that it will take a vehicle a time $\Delta t_i(t^{\text{rs}})$ (h) and distance $\Delta x_i(t^{\text{rs}})$ (km) to reach the effective speed. A vehicle that has not reached the effective speed yet will travel first with its free speed $v_i(t^{\text{rs}})$ (km/h) for a time $\Delta t_i^{\text{free}}(t^{\text{rs}})$ after which it has to start to decelerate – see Figure 2.4 for a graphical representation of these variables. It will then reach the point $x_i(t^{\text{rs}} + \Delta x_i^{\text{free}}(t^{\text{rs}}) + \Delta t_i(t^{\text{rs}}))$ given by:

$$x_i(t^{\text{rs}} + \Delta x_i^{\text{free}}(t^{\text{rs}}) + \Delta t_i(t^{\text{rs}})) = x_i(t^{\text{rs}}) + v_i(t^{\text{rs}})\Delta t_i^{\text{free}} + \Delta x_i(t^{\text{rs}})\,. \tag{2.22}$$

The target trajectory line is then located at:

$$x_i^{\text{target}}(t^{\text{rs}} + \Delta t_i^{\text{free}}(t^{\text{rs}}) + \Delta t_i(t^{\text{rs}})) = x_i^{\text{target}}(t^{\text{rs}}) + v^{\text{eff}}(\Delta t_i^{\text{free}}(t^{\text{rs}}) + \Delta t_i(t^{\text{rs}}))\,. \tag{2.23}$$

Solving these two equations for $\Delta t_i^{\text{free}}(t^{\text{rs}})$ gives:

$$\Delta t_i^{\text{free}}(t^{\text{rs}}) = \frac{x_i(t^{\text{rs}}) + \Delta x_i(t^{\text{rs}}) - x_i^{\text{target}}(t^{\text{rs}}) - v^{\text{eff}}\Delta t_i(t^{\text{rs}})}{v^{\text{eff}} - v_i(t^{\text{rs}})}\,. \tag{2.24}$$

During this time, the vehicle has traveled $\Delta x_i^{\text{free}}(t^{\text{rs}})$ (km) given by:

$$\Delta x_i^{\text{free}}(t^{\text{rs}}) = v_i(t^{\text{rs}})\Delta t_i^{\text{free}}(t^{\text{rs}})\,. \tag{2.25}$$

Vehicles for which it holds that $\Delta t_i^{\text{free}}(t^{\text{rs}}) \leq T^{\text{rs}}$ will have to start decelerating for stabilization during the current sampling time step.

In a real-world implementation it is not necessary (and may not be possible) to test all vehicles upstream of a speed limited vehicle whether it needs to slow down during the current time step as well. However, it is not always sufficient to find the first upstream vehicle i^{last} that does not have to slow down because there may be another vehicle upstream of it that is traveling faster and needs to slow down earlier than or within the same time step as vehicle i^{last}.

Consider the worst case where a vehicle is traveling directly upstream of vehicle i with the maximum speed $v^{\max}$ (km/h). The maximum time $t^{\max}$ that this vehicle will have to decelerate to reach the effective speed is

$$t^{\max} = \frac{v^{\text{eff}} - v^{\max}}{a^{\text{T}}}\,, \tag{2.26}$$

and the distance traveled during deceleration is

$$d^{\max} = \frac{t^{\max}(v^{\text{eff}} + v^{\max})}{2}\,. \tag{2.27}$$

Given this, the time $\Delta t_i^{\text{free,min}}(t^{\text{rs}})$ (h) when this vehicle should start to decelerate is given by:

$$\Delta t_i^{\text{free,min}}(t^{\text{rs}}) = \frac{x_i(t^{\text{rs}}) + \Delta d^{\max} - x_i^{\text{target}}(t^{\text{rs}}) + d^{\text{headway}} - v^{\text{eff}}\Delta t^{\max}}{v^{\text{eff}} - v^{\max}}\,. \tag{2.28}$$

If it is the case that $\Delta t_i^{\text{free,min}}(t^{\text{rs}}) < T^{\text{rs}}$ the possibility exists that there is a vehicle upstream that needs to decelerate earlier than vehicle i. However, if it holds that $\Delta t_i^{\text{free}}(t^{\text{rs}}) > T^{\text{rs}}$, and $\Delta t_i^{\text{free,min}}(t^{\text{rs}}) > T^{\text{rs}}$, this is the last vehicle i^{last} that should be speed-limited for stabilization.

The speed-limits are communicated to the vehicles using the S-tail line $L^{\text{S-tail}}(k^{\text{rs}})$. The S-tail line is defined as the set of lines connecting the points defined by the times $t_i^{\text{S}}(t^{\text{rs}}) = t^{\text{rs}} + \Delta t_i^{\text{free}}(t^{\text{rs}})$ when and locations $x_i^{\text{S}}(t^{\text{rs}}) = x_i(t^{\text{rs}}) + \Delta x_i^{\text{free}}(t^{\text{rs}})$ where the vehicles have to start decelerating for all $i^{\text{us}} \leq i \leq i^{\text{last}}$:

$$L^{\text{S-tail}}(k^{\text{rs}}) = \{(t_i^{\text{S}}(t^{\text{rs}}), x_i^{\text{S}}(t^{\text{rs}}))\}\,\forall\, i^{\text{us}} \leq i \leq i^{\text{last}}\,. \tag{2.29}$$

2.3.4 Step IV: Speed limit release

Since the speed in the stabilization area is constant and the density is homogeneous, the speed limits can be released along a straight line in the time-space plane, from downstream to upstream, similarly to SPECIALIST. Let us call this line the S-head line $L^{\text{S-head}}(k^{\text{rs}})$:

$$L^{\text{S-head}}(k^{\text{rs}}) = x^{\text{S-head,start}}, x^{\text{S-head,start}}, v^{\text{S-head}}\,, \tag{2.30}$$

where the position $x^{\text{S-head,start}}$ (km) and time $x^{\text{S-head,start}}$ (h) define the place where the jam resolved, and the speed $v^{\text{S-head}}$ (km/h) defines the slope of the S-head line.

After releasing the speed limit along the S-head line, the vehicles will accelerate to a free-flow state with a lower density than in the stabilization area. The S-head line should start when the jam is resolved and start at the location where the last vehicle that was in the jam, leaves the jam (corresponding to point D in Figure 2.1, and should apply for all vehicles in the stabilization area, including the ones that have started to decelerate toward the stabilization area.

The slope of the S-head line is a tuning variable, and the more steep it is (more negative) the higher the resulting flow will be.

2.3.5 The target following distance

As stated in Section 2.3, the approach to stabilization of the traffic results in an average density of $\rho^{[4]} = 1/d^{\text{headway}}$ despite the variations in deceleration. To see this, note that the density is defined as:

$$\rho^{[4]} = \lim_{i \to \infty} \frac{N}{L}\,, \tag{2.31}$$

where $N = i - j^{\text{ref}}$ (veh) is the number of vehicles between vehicle i and the downstream vehicle j^{ref} in a stretch of the freeway with length L. Due to variations in the deceleration of traffic, the realized target trajectory $x_i^{\text{target}}(t)$ of vehicle i might deviate some distance d_i^{error} (km) from its target trajectory:

$$x_i^{\text{target}}(t) = x_i^{\text{target}}(t) + d_i^{\text{error}}\,. \tag{2.32}$$

When a vehicle i is N vehicles upstream of the vehicle j^{ref} that started the target trajectory, the distance L between these vehicles is given by:

$$L = N d^{\text{headway}} + d_i^{\text{error}} + d_{j^{\text{ref}}}^{\text{error}}\,. \tag{2.33}$$

Thus, the realized density is given by:

$$\rho^{[4]} = \frac{N}{N d^{\text{headway}} + d_i^{\text{error}} + d_{j^{\text{ref}}}^{\text{error}}}\,, \tag{2.34}$$

which converges to $\rho^{[4]} = 1/d^{\text{headway}}$ for large i:

$$\rho^{[4]} = \lim_{i \to \text{inf}} \frac{N}{N d^{\text{headway}} + d_i^{\text{error}} + d_{j^{\text{ref}}}^{\text{error}}} = \frac{1}{d^{\text{headway}}}\,, \tag{2.35}$$

if d_i^{error} and $d_{j^{\text{ref}}}^{\text{error}}$ are bounded.

2.4 Algorithmic formulation

The previous chapters explained how individual vehicles can detect a jam wave and when vehicles should be speed limited in order to resolve it. This chapter presents how

this behavior can be implemented in a cooperative system. The idea of this algorithm is that vehicles send their current speed, position, and jam detection information to the roadside. The jam detection information is called the detection mode and is described in Section 2.4.1. Based on this information, the roadside system then determines the desired behavior of vehicles on different segments of the freeway. This is done by instructing driving modes which are detailed in Section 2.4.2.

2.4.1 Detection modes

The detection modes are used to detect the congestion and communicate this with the road-side system. The detection modes can be: 1) Detection mode J, jam detected, and 2) Detection mode F, free-flow detected. The detection modes are determined in individual the vehicles according to (2.1)–(2.3). Note that the roadside system receives the position, speed, and detection mode of every vehicle. Using this information, the roadside system can detect the head of the congestion.

2.4.2 Driving modes

The road-side algorithm instructs the driving modes to the vehicles. Three driving modes exist, namely: 1) Mode A, autonomous driving, 2) Mode R, jam resolution, and 3) Mode S, stabilization. In each mode a vehicle has a different role and is controlled according to another control regime. In the following it is described what the conditions are to be in a certain mode, what control rules determine the vehicle's car-following behavior. In all cases it holds that the specified car-following rules should be applied such that the autonomous car-following rules may always override them if it is necessary to ensure safety.

Mode A: autonomous driving

By default, the vehicles drive in mode A, which means that they drive according to their own car-following rules. Autonomous in this context does not mean driverless, or fully automated, but that there is no intervention from the system in the default driving behavior of the vehicle or driver.

Mode R: vehicles that resolve the jam

The vehicles in mode R are the vehicles directly upstream of the jam that have to be slowed down to resolve the jam according to (2.11). These vehicles have therefore a target maximum speed v^{eff}. In general, these vehicles will enter the jam at some point and will have to reduce their speed using their own their own car-following strategies. However, before that, while driving in the upstream free-flow area, they will be able

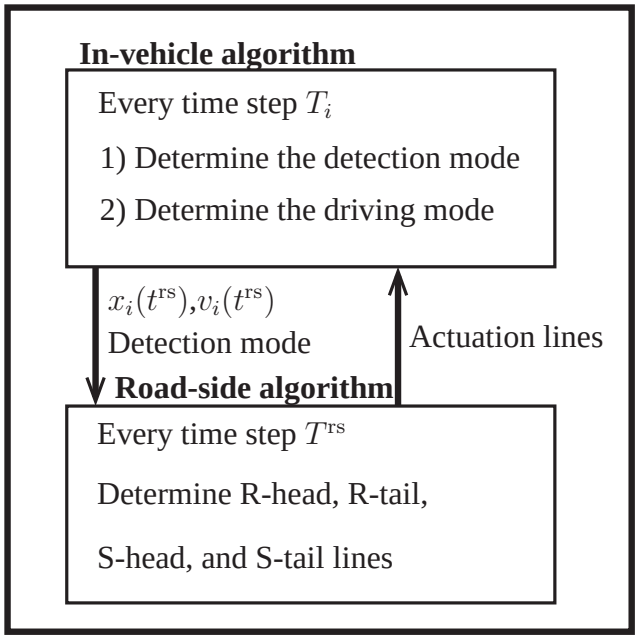


Figure 2.5: Overview of the algorithm.

to travel at speed v^{eff}, and their following distances will be typically too large for the target speed (larger than necessary), since these following distances equal on the average the following distances in free-flow. These free-flow following distances were even large enough for a stable traffic flow in combination with the (higher) free-flow speed, so the string of vehicles in mode R should be stable too.

Mode S: vehicles that stabilize the traffic flow

The vehicles in the stabilization area, created upstream of the vehicles in mode R, are in mode S. These vehicles have a target maximum speed v^{eff}. Which vehicles are in mode S is determined according to the procedure described in Section 2.3.3. In this case, the vehicles are only instructed to reduce their speed in order to ensure compatibility with all types of in-vehicle systems.

2.4.3 Algorithm

The COSCAL v1 algorithm can be sub-divided into an in-vehicle part and a road-side part which are described here. The functionality is illustrated in Figure 2.5.

In-vehicle algorithm

The in-vehicle algorithm has a sampling time step of T_i. At every time step the in-vehicle algorithm receives the position and speed measurements. Next, it has two tasks to complete.

First of all, it determines the jam-state $j_i(k_i)$ using (2.1)–(2.3). When this jam-state is 1 the detection mode is J, otherwise it is F. This detection mode, and the position, lane, and speed of the vehicle are communicated every T^{rs} seconds to the road-side.

Secondly, the in-vehicle algorithm receives every T^{rs} seconds the boundary lines. The in-vehicle algorithm translates these lines into the current locations $x^{\mathrm{R-head}}(k_i)$ (km), $x^{\mathrm{R-tail}}(k_i)$ (km), $x^{\mathrm{S-head}}(k_i)$ (km), and $x^{\mathrm{S-tail}}(k_i)$ (km) of the boundaries. Based on the current position of the vehicle and these locations the vehicle determines its driving mode according to the following logic:

- **Mode A:** a vehicle is by default in to mode A unless it is in mode R or S;
- **Mode R:** if a vehicle is downstream of the R-tail line and upstream of the R-head line, i.e., $x^{\mathrm{R-head}} \leq x_i(k_i) \leq x^{\mathrm{R-tail}}$, it is in mode R;
- **Mode S:** if a vehicle is downstream of the S-tail line and upstream of the R-tail line or S-head line, it is in mode S. Note that the S-tail line can move backward and forward over time. This implies that a vehicle can be downstream of several locations $x^{\mathrm{S-tail}}(k_i)$ at time step k_i. A vehicle is in mode S if it is downstream of an uneven number of locations $x^{\mathrm{S-tail}}(k_i)$.

Road-side algorithm

The road-side algorithm has a sampling time step of T^{rs}. During a sampling time step it receives the locations $x_i(k^{\mathrm{rs}})$, speeds $v_i(k^{\mathrm{rs}})$ and detection modes of all the vehicles on the freeway. Based on this data, the roadside system determines the boundaries between the different driving mode areas for the coming sampling-time step T^{rs}. These boundaries are:

- The R-head line, which exists when congestion is present on the freeway at time $k^{\mathrm{rs}}T^{\mathrm{rs}}$. This line follows the head of congestion.
- The R-tail line, which exists when congestion is present on the freeway at time $k^{\mathrm{rs}}T^{\mathrm{rs}}$ it contains the tail of the speed limited area that is needed to resolve a jam which follows from (2.11). Vehicles in-between the R-head and R-tail line are in mode R.
- The S-tail lines exist when there are speed-limits active. They are the upstream ends of the stabilization areas per lane. The S-tail lines should be determined following the reasoning presented in Section 2.3.3. Vehicles between the S-head or R-tail and an S-tail line are in mode S.
- The S-head line exists when there is an S-tail line, and there is no congestion. It is defined following the reasoning as detailed in Section 2.3.4.

Every road-side sampling time $k^{\mathrm{rs}}T^{\mathrm{rs}}$ the roadside communicates the lines to the vehicles on the freeway.

2.5 Simulation

The developed algorithm is evaluated using the microscopic traffic flow simulator VISSIM 5.40 and Matlab R2015a. The objective of the evaluation is to assess the qualitative behavior – in the sense that the algorithm is able to resolve a jam wave and to stabilize traffic – and to test whether this can result in throughput improvements. Two case studies are carried out, namely, 1) the application of COSCAL v1 to a single lane freeway without driving behavior differences between vehicles, and 2) the application of COSCAL v1 to a two lane freeway where driving behavior differences between vehicles are allowed.

2.5.1 Evaluation I: a single lane freeway

The first case study consisted of a single lane freeway network and no driving behavior differences between vehicles. This simplified case study was selected in order to obtain full control over the experiment. Also, it prevents undesirable effects, such as, a moving bottleneck that is caused by a slow driving vehicle that cannot be overtaken. A one lane freeway of 5 km long was implemented in Vissim. A demand was created by generating an identical vehicle every 1.4 s while skipping occasionally a vehicle, for testing the homogenizing effect of the stabilization mode. This resulted in an inflow of 2323 veh/h. The vehicles had a uniform desired speed of 120 km/h in mode A, and of 80 km/h in modes R and S (assuming that 60 km/h is displayed). It is assumed that the penetration rate is 100% and thus all vehicles receive the instructions and comply with the instructions (in the sense that they all drive 80 km/h if 60 km/h follows from the instructed mode).

A period of 650 seconds was simulated for a jam wave scenario. The jam wave was created by artificially lowering the first vehicle's desired speed to 20 km/h between the 80th and 115th second. The sampling time of the in-vehicle algorithm was set to the simulation time step (0.2s), and the sampling time step of the roadside controller was set to 5 seconds.

The Wiedemann 99 model which is implemented in VISSIM was used to model the driving behavior. For the reproducibility of the experiments but without going in too much detail, the parameters that were changed from the default settings are reported here: the number of vehicles observed ahead: 3, standstill distance: 1.5 m, headway time: 0.9 s, 'following' variation: 4.0 m, threshold for entering 'following': -8.0 m, negative 'following' threshold: -0.35, positive 'following' threshold: 0.35, speed dependency of oscillation: 11.44, oscillation acceleration: 0.25 m/s^2, standstill acceleration: 1.0 m/s^2, and acceleration at 80 km/h:1.50 m/s^2.

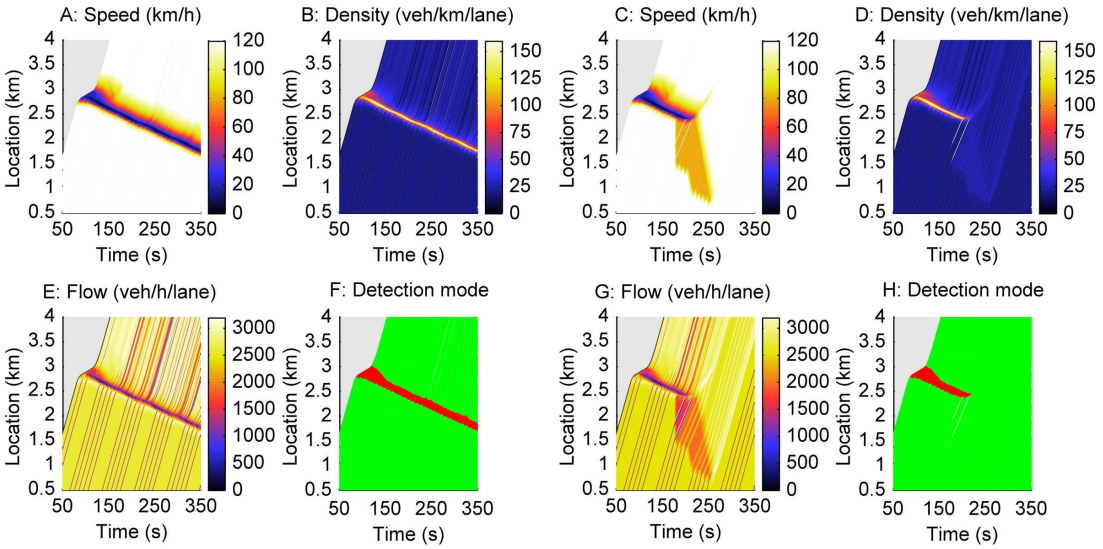


Figure 2.6: The vehicle trajectories per lane for the uncontrolled – A, B, E, and F – and controlled – C, D, G, and H – scenario I. The trajectories are colored according to speed, density and flow in the corresponding sub-figures. In plot F, the figures are colored green when they are in detection mode F, and red when they are in detection mode J. In plot H the trajectories are colored according the driving mode with A: green, R: blue, and S: orange.

Uncontrolled case

The resulting jam can be seen in Figure 2.6. The figure shows the speed, density, flow and the detection modes along each vehicle trajectory. Due to the jam wave scenario, the jam grows when the first vehicle's desired speed is limited to 20 km/h, needs some time to set to a steady state, and finally it propagates upstream with a speed of approximately -18.5 km/h. The queue discharge rate is measured as 2350 veh/h, implying a capacity drop of 23%, since, the freeway capacity was measured as 3000 veh/h. The gaps that are created due to the skipping of some vehicles during the vehicle generation are clearly visible.

Controlled case

For the controlled case, the algorithm was roughly tuned, but even the initial tuning led to acceptable behavior. The parameter settings of the algorithm can be found in Table 2.1.

The control was started after 175 seconds when the jam was fully formed. The vehicle trajectories for the controlled case are shown in Figures 2.6. Several observations can be made from these plots. First of all, it can be observed that the jam wave is resolved around $t = 220$ s, indicating that the algorithm is capable of resolving a jam wave. Secondly, the structure of the algorithm is similar to the SPECIALIST algorithm. Initially, speed-limits are imposed over a stretch of approximately 1 km resulting in a low flow that resolves the jam wave. Upstream of these vehicles the

Table 2.1: The parameter settings used in the evaluations.

Variable	**Value I**	**Value II**
v^{th}	50 km/h	50 km/h
z^{jam}, z^{ff}	0.0417, -0.0417 km	0.0417, -0.0417 km
q^{head}	2800 veh/h	4200 veh/h
v^{head}	-18.5 km/h	-16.5 km/h
v^{eff}	80 km/h	60 km/h
d^{headway}	1/25 veh/km/lane	1/27.5 veh/km/lane
γ	0.5	0.5
a^{T}	-3 m/s^2	-2 m/s^2
v^{max}	130 km/h	130 km/h
$v^{\text{S-head}}$	-180 km/h	-100 km/h
$\epsilon^{v^{\text{eff}}}$	5 km/h	5 km/h

speed-limited area is gradually increased in order to stabilize traffic. When the jam is resolved, speed-limits are released along a straight line causing a high outflow. An important difference is that the COSCAL v1 algorithm allows the speed-limited area to be adjusted over time due to the feedback structure.

The influence of the feedback can be observed at the tail of the blue area containing vehicles in mode R, and at the tail of the orange area containing vehicles in mode S. Interestingly enough, it can be observed that initially – at time 175 s – the algorithm finds the jam resolving vehicle correctly. Later on, the algorithm moves the tail of the blue area upstream and downstream resulting in an overestimation of the required vehicles at the time instance when the jam resolves. The reason why this happens is that the speed with which the jam head propagates upstream decreases when the jam starts to resolve. This is due to the driving behavior created by VISSIM and it is uncertain whether it is realistic. It can also be observed that the speed with which the orange area is moved upstream changes in such a way that the gaps are closed. In this way the density is homogenized.

The following quantitative results were found. The total time spent (TTS) from time 165 s to 650 s of all the vehicles on the freeway in the uncontrolled situation was 14.33 veh·h and in the controlled situation this was 13.28 veh·h implying a gain of 7.3%. The reason why the TTS is improved is that the outflow of the freeway is increased after the jam is resolved as can be observed in Figure 2.7 A. The outflow is higher, since, the flow downstream of the stabilization area is higher than the flow downstream of the jam wave.

Figure 2.7 B shows the density in the stabilization area over time. It can be observed that from time 200 s to 240 s the density is close to the desired density of 25 veh/km. However, when the stabilization area is just created or almost resolved, it is small so the density is not close to the desired density. On average the density is 24.4 veh/km with a standard deviation of 2.4 veh/km. The peaks that can be observed in the density

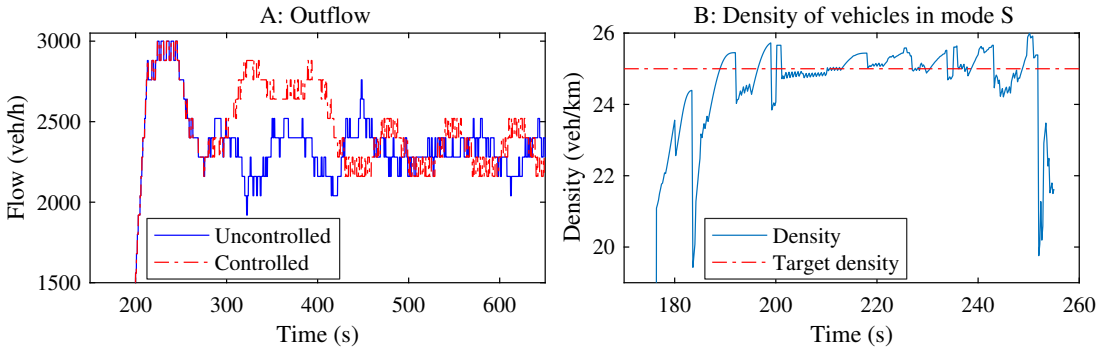


Figure 2.7: A: Comparison of the freeway outflow in the controlled and uncontrolled situation I. B: Density of vehicles in mode S in situation I.

between time 200 s and 240 s are caused by the gaps that were created in the inflow. It takes some time for the algorithm to fill these gaps but it can be observed that the algorithm is able to restore the density to the desired density.

2.5.2 Evaluation II: a two-lane freeway

The objective of the second evaluation was to test the controller performance when applied to a two-lane freeway with driving behavior differences between vehicles. This case study was selected to test the performance of the algorithm in a more realistic set-up. An important difference with the single lane case study is that vehicles can change lanes, and that the traffic flow characteristics, e.g. the location of the jam-head can differ per lane. In this implementation the algorithm is not adjusted to include these effects. Thus, when applying the COSCAL v1 algorithm it assumes that all the vehicles drive in the same lane, and that the traffic flow characteristics are identical per lane. In Section 2.6 the extensions of the algorithm that have to be investigated when dealing with multi-lane freeways are discussed.

A 2 lane freeway of 7.5 km long was implemented in Vissim and speed differences between vehicles were allowed. A constant demand of 3550 veh/h was applied to the freeway and a simulation time of 1800 seconds was considered. A jam wave was created by slowing down vehicles between location 6.9 km and 7.4 km to 0 km/h from time 250 s to 350 s. The evaluation was repeated ten times for different random seeds, namely random seeds 1 to 10.

The Wiedemann 99 model which is implemented in VISSIM was used to model the driving behavior with the following settings: the number of vehicles observed ahead: 3, standstill distance: 3.5 m, headway time: 0.7 s, ‘following’ variation: 6.0 m, threshold for entering ‘following’: -8.0 m, negative ‘following’ threshold: -0.10, positive ‘following’ threshold: 0.10, speed dependency of oscillation: 6.00, oscillation acceleration: 0.25 m/s^2, standstill acceleration: 0.5 m/s^2, and acceleration at 80 km/h:1.50 m/s^2

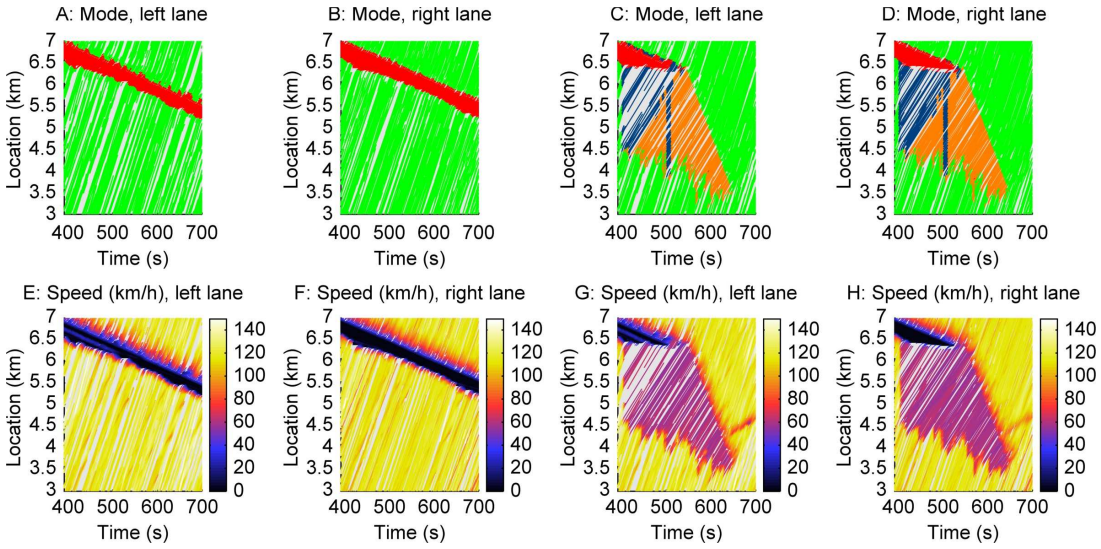


Figure 2.8: The vehicle trajectories per lane for the uncontrolled – A, B, E, and F – and controlled – C, D, G, and H – scenario II. The trajectories are colored according to: (A, B) detection mode with green detection mode F, and red detection mode J, (C, D) driving mode with A: green, R: blue, and S: orange, and (E, F, G, H) speed.

Uncontrolled case

The uncontrolled situation is shown in Figure 2.8 for random seed 2. The detection mode (top) and speed (bottom) in the left and right lane are shown in this figure. The queue discharge rate is measured as 3550 veh/h implying a capacity drop of approximately 5.3%, since, the freeway capacity was approximated as 3750 veh/h. The TTS for all the different random seeds from time 400s to time 800 s is 130.7 veh·h with a standard deviation of 9.5 veh·h.

Controlled case

The control was started after 400 seconds when the jam wave was fully formed. The tuning variables are shown in Table 2.1. Some small changes to the tuning variables used for the first evaluation set were made, namely, the flow over the jam head was set to 4200 veh/h, the speed of the jam head was set to -16.5 km/h, and the target following distance was set to 1/55 veh/km, the deceleration was set to -2m/s^2, and the S-head line speed was set to -100 km/h.

Figure 2.8 shows the trajectories plots for the controlled situation for random seed 2. From this figure it can be observed that the jam wave is successfully resolved. Also, a clear difference can be observed in the jam resolution (blue) area between the left and right lane, clearly vehicles move to the right lane because of the lower flow and speed in this area and in the jam wave the vehicles move back to the left lane. Also, in the stabilization are (orange area) less vehicles seem to be present in the left lane.

Figure 2.9 shows the density in the stabilization area in the left (top) and right (middle)

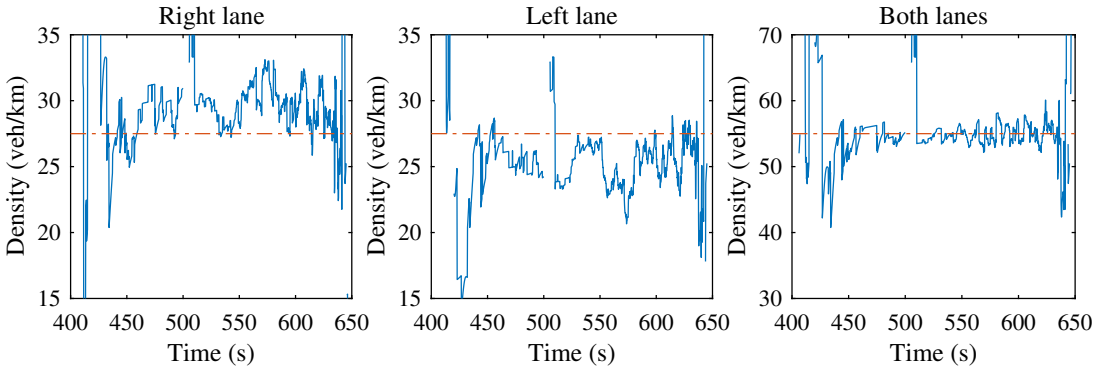


Figure 2.9: The density of vehicles in mode S over time in situation II in the different lanes. The dashed dotted line indicates the target density.

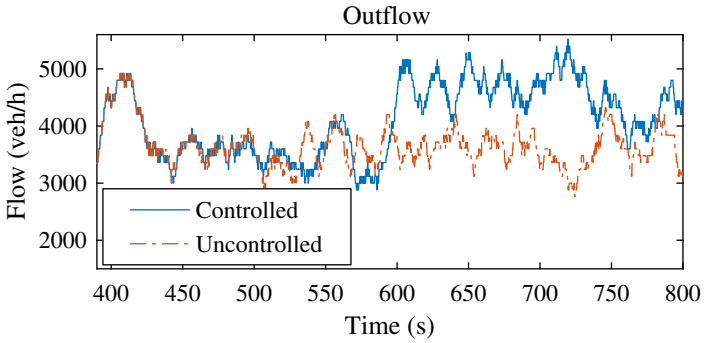


Figure 2.10: Comparison of the freeway outflow in the controlled and uncontrolled situation II.

lane over time. From these plots it can be observed that the density in the left lane is lower than the target density and in the right lane lower. The bottom plot in Figure 2.9 shows that these effects compensate each other resulting in a density in the stabilization area that is near the target density.

Figure 2.10 compares the freeway outflow for the controlled and uncontrolled situation with random seed 2. Similar as in Figure 2.10, the outflow increases after some time. This time – around 600 s – corresponds with the time when the flow out of the stabilization area reaches the downstream end of the freeway.

The following quantitative results were found. The TTS for all the different random seeds from time 400 s to time 800 s was 108.0 veh·h with a standard deviation of 6.1 veh· h indicating an average TTS gain of 17.3%. The average density in the stabilization area on both lanes for the different random seeds was 53.5 veh/km/lane with a standard deviation of 1.4 veh/km. The average density on the right lane was 27.8 veh/km/lane with a standard deviation of 1.3 veh/km/lane and on the left lane it was 24.3 veh/km/lane with a standard deviation of 0.4 veh/km/lane. This indicates that the speed-limited vehicles prefer to drive on the right lane causing a density difference on both lanes which on the average results in a density slightly lower than the target density of 27.5 veh/km/lane.

2.5.3 Concluding remarks on the evaluation

The cooperative speed control algorithm works according to expectations. It is interesting to see that the resulting control scheme is very similar to the SPECIALIST scheme, and comes up as an emergent behavior due to the various mode switching rules and the corresponding calculations, which are very different from the approach used in SPECIALIST. A clear difference is that the edges of the areas with the different modes are not straight, which is expected, and is due to the feedback structure of the cooperative algorithm. The algorithm is able to improve the TTS and delay time.

2.6 Discussion

The algorithm presented in this paper has been developed assuming an ideal situation. This ideal situation consisted of a freeway with a 100% penetration rate of cooperative vehicles. In this section the effect of relaxing the assumptions will be discussed.

First of all, when the algorithm has to deal with lower penetration rates, adaptations have to be made to the detection, and actuation parts of the algorithm. It is expected that in those cases, the algorithm has to be able to deal with both infrastructure-based systems, such as inductive loop detectors, variable message signs, and in-vehicle systems. A lower penetration rate will decrease the accuracy with which the traffic can be observed. Moreover, it will decrease the actuation freedom, since, some of the vehicles can only be influenced using variable message signs. It is expected that a lower penetration rate will reduce the performance of the algorithm. However, the performance is expected to be the same or higher than the performance of the SPECIALIST algorithm. See [Mahajan et al., 2015] for the COSCAL v2 algorithm that is designed to deal with low penetration rates of cooperative vehicles and infrastructure based systems.

Secondly, when on-ramps and off-ramps are included in the algorithm this implies that extra incoming or leaving flow have to be considered when determining the vehicles that should be speed-limited for jam resolution or stabilization. This can influence the length of the jam resolution area. Also, upstream of an on-ramp the density in the stabilization area might be chosen a bit lower in order to accommodate the on-ramp flow. An approach to include a metered on-ramp in the COSCAL v1 algorithm is detailed in [van de Weg et al., 2014a].

Thirdly, the algorithm has been designed for single lane traffic. Although the evaluations suggest that the algorithm can also be effective on a multi-lane freeway, this extension requires further investigation. Research directions are: 1) to study whether the algorithm needs to take traffic flow differences – such as the jam wave characteristics or speed differences – between lanes into account, 2) to study the impact of lane changes on the algorithm, and 3) to determine whether the algorithm should include lane change advice.

Apart from these assumptions, the evaluations that were carried out here have their limitations as well. First of all, the tuning of driving characteristics in Vissim is not easy and the selected parameters do not represent a realistic situation. However, the set-up is sufficient to reach the evaluation objectives. Secondly, the differences in driving behavior and speed can result in unexpected behavior. The algorithm should be adequately tuned. For instance, speed-limited vehicles drive with a range of speeds resulting on the average in the effective speed. When a vehicle prefers to drive with a very low speed, it should not start detecting a jam. Similarly, the algorithm should not get distorted when a vehicle that should drive with the effective speed chooses to drive too fast.

2.7 Conclusion

This paper presented a cooperative speed control algorithm – called COSCAL v1 – that is able to resolve a jam wave. To this end, first the theory to detect a jam wave and determine which vehicles should be speed limited to resolve it was introduced. This theory was developed for an ideal situation assuming a 100% penetration rate of cooperative vehicles. A vehicle to infrastructure communication set-up was adopted in order to realize fast communication. Special attention has been paid to respect the privacy of the users. Also, the system has been developed in such a way that it can deal with in-vehicle speed limits as well as with directly influencing the speed of vehicles. The roadside system imposes the speed-limits by communicating the boundaries between areas in which drivers have different roles to resolve the jam wave. The in-vehicle system is used to detect whether a vehicle is in the jam, and the vehicles send their position, speed, and detection mode to the road-side. It has been shown that by stabilizing the traffic a certain desired density can be realized.

Simulations were carried out using the microscopic simulation software Vissim 5.40 to test the algorithm. First the algorithm was tested for a single lane freeway without driving behavior differences between vehicles in order to test the working of the algorithm. It was found that the algorithm is able to resolve a jam wave. Additionally, it was observed that the structure of the algorithm is, qualitatively, similar to the SPECIALIST algorithm. Although the boundaries of the COSCAL v1 scheme fluctuate more due to the feed-back structure. It was also found that the algorithm is able to improve the TTS with 7.3%. Secondly, the algorithm was applied to a two lane freeway and differences in driving behavior were allowed. It was found that for ten different random seeds the algorithm was able to resolve a jam wave and improve throughput by 17.3% on average, indicating that the algorithm can be applied to more realistic situations although further research has to study the extension of the algorithm to multi-lane freeways.

Further research can be carried out to relax some of the assumptions which were made in this paper. Also, the approach has the potential to be applied to different congestion types or to prevent congestion, which will be part of further research. Finally, the

approach can potentially be extended to deal with more complex traffic networks in which, for instance, multiple on-ramps and off-ramps are present.

Acknowledgment

This work is part of the research programme ‘The Application of Operations Research in Urban Transport’, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

Chapter 3

Efficient MPC for freeway throughput improvement by parameterization of ALINEA and a speed-limited area

In this chapter a computationally efficient approach is developed for the coordination of flows between different freeway network elements. This chapter is based on the following paper that is currently under review:

G.S. van de Weg, A. Hegyi, B. De Schutter, and S.P. Hoogendoorn, Efficient MPC for freeway throughput improvement by parameterization of ALINEA and a speed-limited area. *Transactions on Intelligent Transportation Systems*, submitted 2017-2-17.

Abstract

Freeway congestion can reduce the freeway throughput due to the capacity drop or due to blocking caused by spillback to upstream ramps. Research has shown that congestion can be reduced by the application of ramp metering and variable speed limits. Model predictive control is a promising strategy for the optimization of the ramp metering rates and variable speed limits to improve the freeway throughput. However, several challenges have to be addressed before it can be applied for the control of freeway traffic. This paper focuses on the challenge of reducing the computation time of MPC strategies for the integration of variable speed limits and ramp metering. This is realized via a parameterized control strategy that optimizes the upstream and downstream boundaries of a speed-limited area and the parameters of the ALINEA ramp metering strategy. Due to the parameterization, the solution space reduces substantially, leading to an improved computation time. More specifically, the number of

optimization variables for the variable speed limit strategy becomes independent of the number of variable message signs, and the number of optimization variables for the ramp metering strategy becomes independent of the prediction horizon. The control strategy is evaluated with a macroscopic model of a two-lane freeway with two on-ramps and off-ramps. It is shown that parameterization realizes improved throughput when compared to a non-parameterized strategy when using the same amount of computation time.

3.1 Introduction

Freeway congestion can reduce the freeway throughput causing societal, economical, and environmental costs. Two main reasons exists why congestion reduces throughput. First of all, congestion causes a capacity drop, i.e., the flow downstream of congestion is lower than the capacity flow that can be achieved under free-flow conditions [Hall and Agyemang-Duah, 1991, Kerner and Rehborn, 1996]. Secondly, congestion can spill back in the upstream direction and cause blocking of traffic bound for off-ramps.

Congestion can be mitigated by dynamic traffic management measures. Two popular dynamic traffic management measures on which this paper focuses are ramp metering (RM) and variable speed limits (VSLs). RM is typically used to limit the number of vehicles that want to enter the freeway from an on-ramp using a traffic light. In this way, the flow into a downstream bottleneck can be reduced so that congestion can be prevented, postponed, or resolved. VSLs are speed limits that can be varied over time and are displayed using variable message signs. VSLs can be used to reduce the speed of freeway traffic and they are typically applied for safety reasons. However, several approaches have been designed to reduce freeway congestion using VSLs. In this paper we study the application of RM and VSLs to improve freeway throughput by reducing congestion with the aim of developing an optimization-based control strategy for the integration of VSLs and RM.

3.1.1 Review of RM and VSL strategies

The development of RM and VSL strategies – i.e., control algorithms – is an active research area. In this brief overview we will discuss several VSL and RM strategies that aim at freeway throughput improvement. We will focus here on discussing the mechanisms in traffic flow exploited by the controllers, the controller properties, and investigate challenges and opportunities for further controller development. After concluding this section, we will review the literature on model predictive control strategies for the integration of RM and VSLs in the next section.

VSL

According to Hegyi et al. [2010], two main categories of VSL strategies for the improvement of freeway throughput exist, namely, the homogenizing types and the flow-limiting types. The idea behind the homogenizing types is that by displaying VSLs that are similar to the average speed of the traffic, speed differences between vehicles will be reduced but no significant reduction of the average speed will result [Smulders, 1990, Van den Hoogen and Smulders, 1994, Kühne, 1991]. In this way, the traffic flow is homogenized, resulting in a reduction of the probability of a traffic breakdown, and thus, leading to an improved freeway throughput. However, while field tests did show a reduction in speed differences, implying a more homogeneous traffic flow, no evidence was found for improved freeway throughput [Van den Hoogen and Smulders, 1994].

The main idea behind VSL strategies of the flow-limiting type is that by imposing VSLs the flow on the freeway can be controlled. Several approaches can be found in the literature that are of the flow-limiting type. Carlson et al. [2011] proposed a VSL strategy called mainstream traffic flow control (MTFC) for controlling freeway traffic entering a bottleneck. This gating strategy adjusts the VSL value at a fixed location upstream of a bottleneck in order to create a controlled congestion upstream of the bottleneck with an outflow that is equal to the bottleneck capacity. Several simulation studies were performed showing improved freeway throughput. Challenges of this approach are that very low VSL values may have to be displayed and that the application of the strategy is limited to specific locations in a road network. Besides that, it is an open question whether low VSL values can reduce the freeway flow sufficiently. For instance, Soriguera et al. [2017] carried out an empirical study into the effect of applying speed limit values as low as 40 km/h at a fixed location that showed that applying low VSL values may even result in a flow increase.

Hegyi et al. [2010] proposed a VSL strategy called SPECIALIST based on shock wave theory against jam waves – i.e., congestion with a length of roughly 1 to 2 km that propagates in the upstream direction of the freeway. The SPECIALIST algorithm detects a jam wave and when it assesses this jam wave as resolvable it first applies a pre-defined VSL value instantaneously over a freeway stretch directly upstream of the jam wave. Next, VSLs are imposed upstream of the speed-limited area to stabilize the traffic flow – by creating a stable combination of speed and density – that is approaching the speed-limited area. This causes a reduction of the flow into the jam wave so that it can resolve without triggering an upstream congestion. After the jam wave is resolved, the traffic in the speed-limited area can be released and a higher freeway flow can be achieved since the capacity drop is no longer present. The density and flow in (and downstream of) the speed-limited area can be controlled by adjusting the speed with which the upstream (and downstream) boundary of the speed-limited area propagates. SPECIALIST was tested on the A12 freeway in the Netherlands and it was found that it is capable of resolving jam waves and stabilizing traffic, resulting in improved freeway throughput [Hegyi et al., 2010]. Recently, Mahajan et al. [2015]

proposed a reformulation of SPECIALIST called COSCAL v2. In contrast to the SPECIALIST algorithm which has a feed-forward structure, the COSCAL v2 algorithm has a feedback structure so that it can better adjust its control action to disturbances.

Chen et al. [2014] proposed an alternative approach to resolve congestion at a bottleneck location. In their approach, VSLs are imposed upstream of the bottleneck first so that the congestion head moves away from the bottleneck and the impact of the capacity drop is decreased. After that, by adjusting the VSL values, the outflow of the speed-limited area is adjusted so that it matches the bottleneck capacity. To the best knowledge of the authors, no simulation studies have been carried out yet with this algorithm.

Recently, Zhang and Ioannou [2016] proposed a VSL control strategy integrated with a lane change control strategy to reduce bottleneck congestion caused by incidents. In their approach, lane change control is used to remove the capacity drop and VSL control is used upstream of the incident location to realize target densities that maximize the bottleneck flow.

RM

Similar to VSL strategies of the flow-limiting type, RM is primarily used to limit the freeway flow. The most well-known RM algorithm is ALINEA [Hadj-Salem et al., 1990]. This feedback control strategy for a single on-ramp uses measurements downstream of the on-ramp and regulates the on-ramp flow with the objective of keeping the freeway flow near its critical density. In this way, congestion caused by excessive on-ramp flows can be prevented or postponed and in this way, the impact of the capacity drop is reduced, resulting in improved freeway throughput. Several other control strategies for single on-ramps exist. Middelham and Taale [2006] discusses a demand-capacity RM strategy that uses upstream freeway flow measurements in order to maximize the freeway flow. Due to its feed-forward nature its performance may deteriorate due to disturbances in the traffic flow. A major challenge of these local RM strategies is that the on-ramp queue may spill back to the upstream urban network. Queue management may help to limit the on-ramp queue but also reduces the time that RM can be effective [Papamichail and Papageorgiou, 2008, Carlson et al., 2014].

Coordination of RM at multiple on-ramps can help to extend the RM time. HERO is an algorithm that coordinates the ALINEA-based RM actions of different on-ramps [Papamichail and Papageorgiou, 2008]. Whenever the queue caused by RM at a downstream on-ramp exceeds a threshold, the upstream RM installation starts an RM algorithm that aims at controlling the upstream queue towards a set-point determined by the downstream on-ramp. This prevents the queue at the downstream on-ramp from exceeding the maximum length and allows a longer RM time. Difficulties of coordination are that there exist time delays between the interactions of on-ramps and that not all traffic of upstream on-ramps might be headed to the bottleneck. Not including these

effects may cause unnecessary delays for traffic that is not headed to the bottleneck, which may not be fair [Kotsialos and Papageorgiou, 2004]. One way to include these effects is by predicting the (near) future impact of the control signal on the system performance. Model-based optimal control approaches are typically suited to include such effects and will be discussed in the next section.

Integrated approaches to RM and VSL

Integrating RM and VSL strategies is expected to lead to further freeway performance improvements. From a control engineering point of view this can be explained by the fact that the control freedom is increased, from a traffic-flow-theoretical point of view this can be explained by the possibility to distribute the flow-limiting task over freeway traffic and on-ramp traffic. Schelling et al. [2011] proposed an extension of SPECIALIST so that it can cope with a metered on-ramp. van de Weg et al. [2014a] extended the in-car algorithm COSCAL v1 – which is similar to SPECIALIST – with RM. Mahajan et al. [2015] extended a macroscopic version of COSCAL v1, named COSCAL v2 with RM. In these approaches, it is computed at what time RM is switched on in order to assist the VSL system that resolves jam waves. These studies show that is it possible to integrate the VSL and RM task to resolve jam wave using limited computation time when considering only a single on-ramp. However, a challenge may be the extension to multiple on-ramps, which may lead to a complex control problem due to the time delays between the effects of different actuators.

Carlson et al. [2014] integrated the MTFC approach with RM. They apply ALINEA RM in order to prevent congestion from forming at the bottleneck location. When the on-ramp is full or when the RM rate is near its minimum allowed rate, MTFC control is switched on in order to prolong the RM time. The authors showed that the approach outperforms non-integrated algorithms and realizes a performance that is near the performance realized with optimal control for a bottleneck scenario simulated using a macroscopic traffic flow model. An advantage of this approach is that it is based on a simple feedback control structure.

Conclusions from the literature

In conclusion, RM and VSLs can both limit the freeway flow. These flow reductions can be used to prevent, postpone, or resolve congestion, resulting in improved freeway throughput, since the impact of the capacity drop is reduced. Various algorithms have been developed for RM and VSLs. These algorithms differ in the traffic-flow-theoretical mechanisms that they exploit and their control-theoretical structure. Studies have shown that integrating RM and VSLs can lead to a better performance when compared to isolated systems. However, the control of multiple RM and VSL gantries is a complex problem due to the time delay in the impact of elements on each other.

3.1.2 Review of model-based optimization strategies for freeway traffic control

A promising approach to account for the time delays of control actions on the network-wide performance is model predictive control (MPC) [Rawlings and Mayne, 2009]. MPC uses a prediction model to predict the state of a process over a period of time – called the prediction window – given the current state, a prediction of the disturbances – i.e., inputs that cannot be controlled –, and a candidate control signal. Based on this prediction the performance of the process is expressed using an objective function. Using an optimization technique the control signal is found that leads to the minimum (or maximum) of the objective function. The first step of the control signal is applied to the process, and at the next time step, when new measurements are available, the control signal is optimized again. This is called the receding horizon principle.

Despite the advantages of MPC there also exist several open problems when it is applied to freeway traffic control as discussed in detail in [Burger et al., 2013]. Some key problems are that an accurate prediction of the traffic demand should be available, that the controller should be able to deal with uncertainties, and that the computation time used by the controller should be short enough for real-time application. In this paper we will focus on reducing the computation time of an MPC strategy.

Several authors have applied MPC to the freeway traffic control problem. Kotsialos et al. [2005] and Hegyi et al. [2005a] used the second-order METANET model as a prediction model to optimize RM and integrated RM and VSL settings respectively. An advantage of using second-order models is that they can model more complex traffic dynamics. However, a major challenge is that the nonlinear optimization problem is computationally hard so that real-time application to large freeway networks is not feasible.

Roughly three main approaches exist to limit the computation time required by an MPC strategy. The first is to use computationally efficient traffic flow models. To this end, Gomes and Horowitz [2006] and Hajiahmadi et al. [2015b] use first-order traffic flow models to formulate linear and mixed integer linear optimization problems respectively. The disadvantage of using first-order traffic flow models is that some characteristics of the traffic dynamics may be lost. This may cause a performance loss when applied to a more complex traffic process.

The second strategy is to divide the optimization problem in multiple, possibly overlapping, sub-problems. One such strategy is distributed MPC as in [Frejo and Camacho, 2012]. In such approaches, the freeway network is divided into smaller sub-networks. The sub-problems that need to be solved involve optimization of the sub-network performance and the impact on the total network performance. In some cases this might lead to reduced computation times and similar performance as centralized MPC.

The third strategy is to reduce the number of control parameters that need to be optimized by parameterizing existing control strategies. For instance, Zegeye et al. [2012]

integrated the ALINEA algorithm and a feedback algorithm for VSLs so that only the gains of the feedback strategies had to be optimized. The approach was only applied to cases where the same strategy was used for every actuator type – i.e., VSL or RM – in the network at every time step. Lu et al. [2011] first designed the VSL signal after which the RM rates could be computed using a linear optimization problem. Recently, van de Weg et al. [2015] proposed a parameterization based on SPECIALIST to resolve jam waves using VSLs so that the size of the optimization problem becomes independent of the number of VSL gantries. It is shown using simulations that this approach is able to realize similar performance as the MPC proposed by Hegyi et al. [2005a] in significantly less CPU time while outperforming the approach of Zegeye et al. [2012]. A limitation of the approach of van de Weg et al. [2015] is that it is not yet suited to account for RM and that the performance is only tested in a scenario where throughput is improved by resolving a jam wave.

3.1.3 Research approach and contributions

This paper presents a parameterized MPC strategy for integrated RM and VSLs to improve the freeway throughput. In this way, a better trade-off between the realized throughput improvement and the utilized computation time for integrated optimization of RM and VSL is obtained. The method generalizes the previous work of van de Weg et al. [2015]. Compared to that work, two main contributions are made. First of all, the parameterized VSL approach is extended with a parameterized RM control strategy. Secondly, an extensive qualitative analysis into the controller behavior is carried out when applying the strategy to a jam wave and a bottleneck scenario. Also, the qualitative behavior of the different combinations of RM and VSL is studied. In contrast to the work of Zegeye et al. [2012], per RM installation the RM gain and set-point, and switching times are added to the optimization problem. The switching times are used to change the feedback policy when the traffic situation changes. The parameterization of VSLs and RM rates in METANET is formulated in such a way that the optimization problem can be solved using gradient-based solvers, which are generally faster compared to gradient-free solvers when the problem size is not too large. The third contribution of this paper is to provide insight into the impact of the available computation time budget on the controller performance.

3.2 Controller design

The parameterized MPC strategy proposed in this paper is able to optimize both RM rates and VSL values with the aim of improving the freeway throughput. In the approach proposed in this paper the head and tail of a speed-limited area are parameterized. In this way the number of optimization parameters becomes independent of the freeway length, which would be the case when using non-parameterized optimization

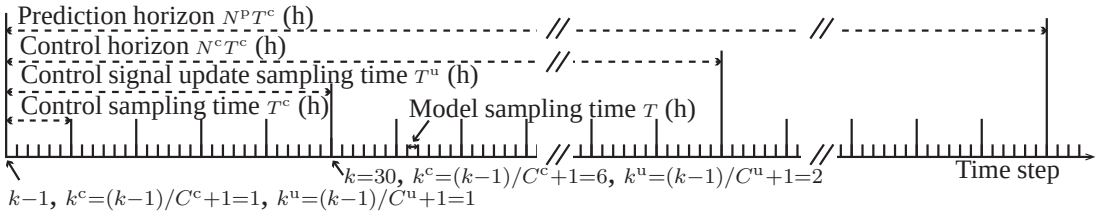


Figure 3.1: Overview of the timing used in the paper for T is 10 s, T^{c} is 60 s, T^{u} is 300 s.

approaches. Additionally, we optimize the parameters of the ALINEA strategy and we optimize the switching times when the controllers should change the parameters of the ALINEA strategy or when they should switch RM off. In this way, the number of optimization parameters for every RM installation becomes independent of the prediction horizon.

3.2.1 Design considerations

Several design considerations are taken into account when developing the parameterized MPC strategy. Special attention is payed to satisfy the requirements for applying RM or VSLs for freeway traffic control. While the primary objective of this paper is to design a control strategy of which the computation time required by the controller is lower than the controller sampling time, (which is in the range of (several) minutes), some design requirements are taken into account as well, which are also important for the practical applicability of this method, namely:

1. Only a limited number of VSL values can be displayed. For instance, in the Netherlands it is only possible to show 50, 60, 70, 80, 90, and 100 km/h.
2. A VSL or RM system should not cause unsafe situations.
3. An RM system typically causes a queue on the on-ramp. The queue length should be bounded by a maximum value to avoid spillback to the upstream road network.
4. The RM rate is typically bounded by a minimum and maximum value.

Below, first the design considerations the VSLs are introduced, followed by the considerations for implementing RM.

VSL control design considerations

As indicated by van de Weg et al. [2014b], a speed-limited area – as shown in Figure 3.2 A – can be created by imposing VSLs. It follows from shock-wave theory that there is a relation between the slope of the boundaries of the speed-limited area and the

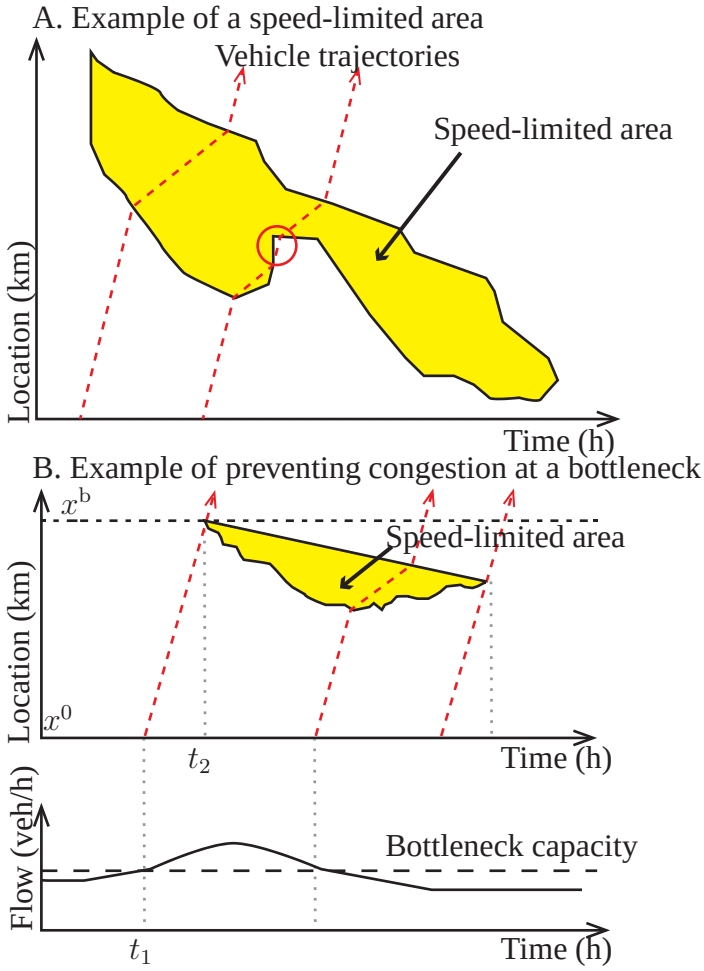


Figure 3.2: A: Example of a speed-limited area that can be used to influence the traffic flow. The red-dashed lines indicate examples of vehicle trajectories. The second vehicle trajectory illustrates a vehicle experiencing a speed limit drop twice – as indicated with the red circle –, which should not occur. B: Top figure: example of a speed-limited area that can be used to prevent congestion at the bottleneck location x^{b}. Bottom figure: the demand entering the freeway at location x^0 [van de Weg et al., 2015].

resulting flow and density downstream of that slope [Hegyi et al., 2010, Lighthill and Whitham, 1955]. If the slope is steeper (more negative) then the resulting density and flow are higher. By adjusting the speed with which the upstream boundary – i.e., the tail – propagates over time, a stable combination of density and flow can be realized in the speed-limited area. Similarly, by adjusting the speed with which the downstream boundary – i.e., the head – propagates over time, the outflow of the speed-limited area can be controlled so that it is just below or at the freeway capacity. SPECIALIST is an example of an algorithm that uses a speed-limited area to resolve a jam wave [Hegyi et al., 2010].

Figure 3.2 B presents an example of using a speed-limited area in order to prevent congestion at a bottleneck. At time t_1 (h) an excess demand – as illustrated in the bottom figure – enters the freeway at location x^0 (km). The time-space plot in the top figure shows that this demand reaches the bottleneck location x^{b} (km) at time t_2 (h). At this time, congestion would appear in a no control situation. However, by imposing

a speed-limited area as illustrated in the top figure, congestion may be prevented.

Several design considerations are taken into account when implementing a speed-limited area. First of all, it is assumed that the value of the speed-limits in the speed-limited area is constant over time. This implies that a segment between two variable message signs is either speed-limited or not. Additionally, it is assumed that only one speed-limited area can be active at a time. Apart from that, the dynamics of the head and tail of the speed-limited area should be such that the individual vehicles can only enter and exit the speed-limited area once. If an individual vehicle observes multiple fluctuations of the speed limits, this can lead to unsafe situations, annoyance, or poor compliance. As an example, the second vehicle in Figure 3.2 A experiences such fluctuations. In order to prevent such behavior, the positions $x^{\mathrm{H,sl}}$ (km) and $x^{\mathrm{T,sl}}$ (km) of respectively the head and the tail of the speed-limited area are allowed to propagate in the downstream direction with a speed that is lower or equal to the effective speed v^{eff}. In the upstream direction they can propagate with any speed.

The speed in the speed-limited area is equal to the effective speed v^{eff} (km/h) corresponding to the imposed VSLs. The effective speed is defined as the speed with which vehicles drive in the speed-limited area which includes possible non-compliance. This can be estimated e.g. from field tests as presented in [Hegyi et al., 2010].

The proposed parameterization reduces the number of optimization variables for VSLs to two per control time step. Note that the number of optimization variables at every control time step used in a nominal MPC strategy is equal to the number of VSL actuators. Hence, the advantage of this parameterization is that the number of optimization variables is reduced, and that the number of optimization variables is independent of the number of VSL actuators.

RM control design considerations

A feedback RM algorithm is used in this paper to control the on-ramp flow that has to satisfy the following properties:

- The RM rate $r_o(k)$ (-) of an origin o should be between the minimum allowed RM rate $r^{\min} \geq 0$ (-) and 1.
- The on-ramp queue length $w_o(k)$ (veh) should not exceed its maximum value $w_o^{\max}$ (veh).

Different RM strategies could be applied depending on the traffic situation. For instance, when preventing congestion at a bottleneck location, the most sensible control strategy would be to control the on-ramp flows in such a way that the flow into a bottleneck is at or just below its capacity. The ALINEA algorithm is specifically designed to

realize this objective. The ALINEA algorithm has the following form [Papageorgiou et al., 1988]:

$$r_o(k+1) = r_o(k) + K_o \frac{\rho_m^{\text{crit}} - \rho_{m,1}(k)}{\rho_m^{\text{crit}}}\,, \tag{3.1}$$

where ρ_m^{crit} (veh/km/lane) is the critical density of the link directly downstream of the on-ramp, and $\rho_{m,1}(k)$ (veh/km/lane) is the current density in the most upstream segment of the downstream link.

When resolving a jam, the flow into the jam should be reduced as much as possible. The standard ALINEA RM algorithm is not suited to realize this, since it tries to fit as much traffic onto the freeway without exceeding the critical density. This can be solved by adapting the set-point $\rho_o^{\text{set}}(k)$ (veh/km/lane) of the ALINEA strategy [Smaragdis et al., 2004, Zegeye et al., 2012]:

$$r_o(k+1) = r_o(k) + K_o \frac{\rho_o^{\text{set}}(k) - \rho_{m,1}(k)}{\rho_o^{\text{set}}(k)}\,. \tag{3.2}$$

Another advantage of including such a set-point is that coordination of on-ramps becomes possible. In the case of a downstream bottleneck or congestion, the set-points of the controllers of different on-ramps can be coordinated in order to distribute the RM task over the RM installations.

Finally, it might be necessary to switch set-points a certain number of times. For instance, when resolving a jam, the preferred strategy might be to reduce the on-ramp inflow as much as possible until the moment when the jam has been resolved and afterwards the freeway flow can be increased to capacity so that the on-ramp outflow can also be increased. These two different tasks require different set-points. Therefore, we propose the following feedback control algorithm:

- Initially, RM is off until switching time $t_{o,1}^{\text{switch}}$ (h).
- From switching time $t_{o,1}^{\text{switch}}$ until switching time $t_{o,2}^{\text{switch}}$ (h), the feedback law (3.2) with feedback gain $K_{o,1}^{\text{s}}$ and set-point $\rho_{o,1}^{\text{set}}$ (veh/km/lane) is used.
- From switching time $t_{o,2}^{\text{switch}}$ until switching time $t_{o,3}^{\text{switch}}$ (h), the feedback law (3.2) with feedback gain $K_{o,2}^{\text{s}}$ and set-point $\rho_{o,2}^{\text{set}}$ (veh/km/lane) is used.
- After time $t_{o,3}^{\text{switch}}$ the RM installation is switched off.

This parameterization requires 5 parameters per RM installation, namely, three switching times, and two set-points. If needed, the approach can be extended by adding more switching time instants or to optimize the feedback gains, which are now manually tuned.

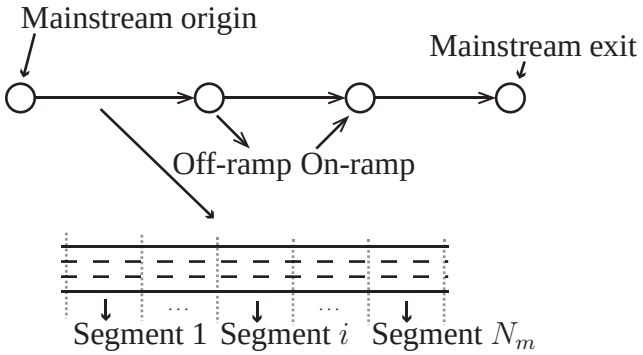


Figure 3.3: Example of the METANET elements used in this paper. A freeway consist of mainstream origins, links, segments, off-ramps, on-ramps, and mainstream exits.

3.2.2 Timing

Before continuing, the timing of the approach is introduced and is illustrated in Figure 3.1. The discrete-time second-order traffic model METANET is used to describe the evolution of the traffic [Kotsialos et al., 2002a]. The time step of the model is indicated with k (-) and the corresponding sampling time with T (h). The time step k (-) refers to the period $\big[Tk,\, T(k+1)\big)$. The control signal sampling time is $T^{\mathrm{c}} = C^{\mathrm{c}}T$ (h) with $C^{\mathrm{c}} \in \mathbb{N}^{+}$ (-), meaning that the value of the control signal can change at time instants $k^{\mathrm{c}}T^{\mathrm{c}}$ (-). The control signal is updated at time instant $k^{\mathrm{u}}T^{\mathrm{u}}$ (s) for which it holds that the control signal update time $T^{\mathrm{u}} = C^{\mathrm{u}}T$ (h) with $C^{\mathrm{u}} \in \mathbb{N}^{+}$ (-). Note that it holds that $t = Tk = T^{\mathrm{c}}k^{\mathrm{c}} = T^{\mathrm{u}}k^{\mathrm{u}}$. The controller predicts the evolution of the traffic from control time step $k^{\mathrm{c}}+1$ until control time step $k^{\mathrm{c}}+N^{\mathrm{p}}$ where N^{p} (-) is the prediction horizon. The control input from control time step k^{c} until control time step $k^{\mathrm{c}}+N^{\mathrm{c}}$ is optimized by the controller where N^{c} (-) is the control horizon and $N^{\mathrm{c}} \leq N^{\mathrm{p}}$. After the control horizon the control signal is taken to be constant.

3.2.3 Traffic flow modelling

An extended version of the METANET model is adopted to predict the evolution of the traffic in the MPC controller. The METANET model presented in [Kotsialos et al., 2002a] along with the extensions proposed in [Hegyi et al., 2005a] is adopted since it provides a detailed description of the traffic dynamics and it can reproduce relevant traffic characteristics such as jam waves and the capacity drop. Note that in the description below only the elements relevant for this paper are discussed. For a full description of the model see [Kotsialos et al., 2002a] and [Hegyi et al., 2005a].

The original METANET model and existing extensions

In the METANET model, a freeway is divided into homogeneous – i.e., having a constant number of lanes, no on-ramps and off-ramps, and constant characteristics – links

m that are connected by nodes [Kotsialos et al., 2002a]. Each link m consists of N_m (-) segments of length L_m (km) with a number of λ_m (-) lanes. The flow $q_{m,i}(k)$ (veh/h), density $\rho_{m,i}(k)$ (veh/km/lane) and speed $v_{m,i}(k)$ (km/h) in a link are updated according to:

$$q_{m,i}(k) = \rho_{m,i}(k) v_{m,i}(k) \lambda_m \,, \tag{3.3}$$

$$\rho_{m,i}(k+1) = \rho_{m,i}(k) + \frac{T}{L_m \lambda_m}\big(q_{m,i-1}(k) - q_{m,i}(k)\big), \tag{3.4}$$

$$\begin{aligned} v_{m,i}(k+1) = v_{m,i}(k) &+ \frac{T}{\tau}\big(V\big(\rho_{m,i}(k)\big) - v_{m,i}(k)\big) \\ &+ \frac{T}{L_m} v_{m,i}(k)\big(v_{m,i-1}(k) - v_{m,i}(k)\big) \\ &- \frac{\eta T}{\tau L_m} \frac{\rho_{m,i+1}(k) - \rho_{m,i}(k)}{\rho_{m,i}(k) + \kappa} \,, \end{aligned} \tag{3.5}$$

In the latter equation, τ and κ are model parameters. The parameter η (-) is set to η^{high} when the downstream density is higher than the density $\rho_{m,i+1}(k)$ in segment i, and it is set to η^{low} when the downstream density is lower. This adjustment is adopted from [Hegyi et al., 2005a] to reproduce the capacity drop. The speed $V\big(\rho_{m,i}(k)\big)$ (km/h) is given as:

$$V\big(\rho_{m,i}(k)\big) = \min\Big[v_m^{\text{free}} \exp\Big(-\frac{1}{a_m}\Big(\frac{\rho_{m,i}(k)}{\rho_m^{\text{crit}}(k)}\Big)^{a_m}\Big), v_{m,i}^{\text{ctrl}}(k)\Big] \,, \tag{3.6}$$

where a_m (-) is a model parameter, the speed v_m^{free} (km/h) is the free-flow speed of link m, and the density ρ_m^{crit} (veh/km) is the critical density, and where the speed $v_{m,i}^{\text{ctrl}}(k)$ (km/h) is the effective speed of the imposed speed limits that is corrected for the compliance of the road-users.

An origin is modeled using a simple queuing model describing the number of vehicles $w_o(k)$ (veh) in the origin queue as a function of the demand $d_o(k)$ (veh) and the outflow $q_o(k)$ (veh/h):

$$w_o(k+1) = w_o(k) + T\big(d_o(k) - q_o(k)\big) \,. \tag{3.7}$$

When an origin acts as the mainstream origin, the outflow is given by:

$$q_o(k) = \min\Big[d_o(k) + \frac{w_o(k)}{T}, q_{\mu,1}^{\lim}(k)\Big] \,, \tag{3.8}$$

where the flow $q_{\mu,1}^{\lim}(k)$ is determined by the traffic condition in the first link and the speed $v_{\mu,1}^{\lim}(k) = \min[v_{\mu,1}^{\text{ctrl}}(k), v_{\mu 1}(k)]$ as follows:

$$q_{\mu,1}^{\lim}(k) = \begin{cases} \lambda_\mu v_{\mu,1}^{\lim}(k) \rho_\mu^{\text{crit}}\Big[-a_\mu \ln\Big(\dfrac{v_{\mu,1}^{\lim}(k)}{v_\mu^{\text{free}}(k)}\Big)^{1/a_\mu}\Big] & \text{if } v_{\mu,1}^{\lim}(k) < V\big(\rho_\mu^{\text{crit}}(k)\big) \\ q_\mu^{\text{cap}} & \text{if } v_{\mu,1}^{\lim}(k) \geq V\big(\rho_\mu^{\text{crit}}(k)\big) \end{cases} \tag{3.9}$$

When an origin acts as a metered on-ramp, the outflow is given by:

$$q_o(k) = \min\left[d_o(k) + \frac{w_o(k)}{T}, Q_0 r_o(k), Q_0 \frac{\rho_m^{\max} - \rho_{m,1}(k)}{\rho_m^{\max} - \rho_m^{\text{crit}}}\right], \tag{3.10}$$

with Q_0 (veh/h) the on-ramp capacity, and $r_o(k) \in [0, 1]$ the RM rate.

In the case that there is an on-ramp upstream of link m, then the term

$$-\frac{\delta T q_o(k) v_{m,1}(k)}{L_m \lambda_m (\rho_{m,1}(k) + \kappa)}, \tag{3.11}$$

is added to (3.5) for the first segment of link m with δ (-) a model parameter.

Finally, when a link m has no leaving link – i.e., it is the most downstream link – the density $\rho_{m,i_m^{\text{last}}+1}$ downstream of the last segment i_m^{last} is equal to:

$$\rho_{m,i_m^{\text{last}}+1} = \max\left[\rho^{\text{DS}}(k), \min[\rho_{m,i_m^{\text{last}}}(k), \rho_m^{\text{crit}}]\right], \tag{3.12}$$

where the density $\rho^{\text{DS}}(k)$ (veh/km/lane) is the destination density, which can be used as a boundary condition to the model.

3.2.4 Extensions for parameterized MPC

This section details extensions that are included in order to use the model for parameterized MPC. These extensions do not affect the dynamic equations of the traffic states but rather the equations that relate the parameterized control signals to the dynamic equations to the control signals. Although the paper focuses on the use of METANET, the extensions may also be used in combination with other macroscopic traffic flow models.

Extension with a speed-limited area

In this paper, the VSLs $v_{m,i}^{\text{ctrl}}(k)$ are determined by the head $x^{\text{H,sl}}(k)$ (km) and tail $x^{\text{T,sl}}(k)$ (km) of the speed-limited area as follows:

$$v_{m,i}^{\text{ctrl}}(k) = \tag{3.13}$$
$$\begin{cases} v^{\text{eff}} & \text{if } x^{\text{H,sl}}(k) > x_{m,i} \text{ and } x^{\text{T,sl}}(k) < x_{m,i} + L_m \text{ and } x^{\text{H,sl}}(k) > x^{\text{T,sl}}(k) \\ v^{\text{free}} & \text{otherwise}, \end{cases}$$

where $x_{m,i}$ (km) is the most upstream location of segment i of link m.

In practice, the speed-limited area can either cover an entire segment or not cover it at all. This implies that the gradient of the objective function is not a continuous function of the location of the speed-limited area. In order to realize a gradient of the VSL signal

that is differentiable everywhere, a parameter $\gamma_{m,i}(k)$ (-) is introduced. The parameter $\gamma_{m,i}(k)$ denotes the fraction of the segment that is covered by speed limits given as:

$$\gamma_{m,i}(k) = \max\left[\frac{L_m - \max[x^{\mathrm{T,sl}}(k) - x_{m,i}, 0] - \max[x_{m,i} + L_{m,i} - x^{\mathrm{H,sl}}(k), 0]}{L_m}, 0\right]. \tag{3.14}$$

In the optimization, the speed $v_{m,i}^{\mathrm{ctrl}}(k)$ in (3.6) is replaced by $\hat{v}_{m,i}^{\mathrm{ctrl}}(k)$ by taking the weighted average of the effective speed v^{eff} and the equilibrium speed $v^{\mathrm{FD}}\big(\rho_{m,i}(k)\big)$:

$$\hat{v}_{m,i}^{\mathrm{ctrl}}(k) = \gamma_{m,i}(k)v^{\mathrm{eff}} + (1 - \gamma_{m,i}(k))v^{\mathrm{FD}}\big(\rho_{m,i}(k)\big)\,. \tag{3.15}$$

Extension with feedback ramp metering

The feedback RM control strategy results in a flow reduction factor $\tilde{r_o}(k)$ (-) that limits the on-ramp flow [Kotsialos et al., 2005]. The overall RM control strategy is as follows: until time $t_{o,1}^{\mathrm{switch}}$ RM is off and the RM rate is equal to 1; this policy is indicated with policy index $i^{\mathrm{p}} = 1$ (-). After that time until time $t_{o,2}^{\mathrm{switch}}$ the ALINEA algorithm is used to meter the on-ramp traffic with the gain $K_{o,2}^{\mathrm{s}}$ to reach the set point $\rho_{o,2}^{\mathrm{set}}$; this corresponds to policy $i^{\mathrm{p}} = 2$. After time $t_{o,2}^{\mathrm{switch}}$ until time $t_{o,3}^{\mathrm{switch}}$ the maximum queue length strategy is used with gain $K_{o,3}^{\mathrm{s}}$ to reach the set-point $\rho_{o,3}^{\mathrm{set}}$; this corresponds to policy $i^{\mathrm{p}} = 3$. After time $t_{o,3}^{\mathrm{switch}}$ the RM rate is switched to 1; this corresponds to policy $i^{\mathrm{p}} = 4$. In total a number of $n^{\mathrm{pol}} = 4$ (-) policies per ramp are available.

The switching time instants $t_{o,i^{\mathrm{p}}}^{\mathrm{switch}}$ are real-valued while the actual model timing is discrete. This leads to a discontinuous gradient. In order to prevent this, the RM rates of the different policies $\tilde{r}_{o,i^{\mathrm{p}}}(k)$ are linearly interpolated giving the potential RM rate $\tilde{r_o}(k)$ (-) when a switching time lies in a time interval:

$$\tilde{r_o}(k) = \sum_{i^{\mathrm{p}}=1}^{n^{\mathrm{pol}}} f_{i^{\mathrm{p}}}^{\mathrm{p}}(k)\tilde{r}_{o,i^{\mathrm{p}}}(k)\,, \tag{3.16}$$

where the RM rates $\tilde{r}_{o,i^{\mathrm{p}}}(k)$ (-) of the policies i^{p} are given as:

$$\tilde{r}_{o,i^{\mathrm{p}}}(k) = \begin{cases} 1 & \text{if } i^{\mathrm{p}} = 1 \text{ or } i^{\mathrm{p}} = n^{\mathrm{pol}} \\ \max\left(\min\left(\tilde{r_o}(k-1) + K_{o,i^{\mathrm{p}}}^{\mathrm{s}}\dfrac{\rho_{o,i^{\mathrm{p}}}^{\mathrm{set}} - \rho_{m,1}(k-1)}{\rho_{o,i^{\mathrm{p}}}^{\mathrm{set}}}, 1\right), 0\right) & \text{otherwise,} \end{cases} \tag{3.17}$$

and the fraction $f_{i^{\mathrm{p}}}^{\mathrm{p}}(k)$ represents the fraction of the time step that is covered by policy i^{p} and which is computed using :

$$f_{i^{\mathrm{p}}}^{\mathrm{p}}(k) = \begin{cases} \dfrac{\max[0, T + \min[t_{o,i^{\mathrm{p}}}^{\mathrm{switch}} - kT]]}{T} & \text{if } i^{\mathrm{p}} = 1 \\ \dfrac{\max[0, T - \max[t_{o,i^{\mathrm{p}}-1}^{\mathrm{switch}} - (k-1)T, 0]}{T} & \text{if } i^{\mathrm{p}} = n^{\mathrm{pol}} \\ \dfrac{\max[0, T - \max[t_{o,i^{\mathrm{p}}-1}^{\mathrm{switch}} - (k-1)T, 0] + \min[t_{o,i^{\mathrm{p}}}^{\mathrm{switch}} - kT]]}{T} & \text{otherwise.} \end{cases} \tag{3.18}$$

The next step is translating the RM rate $\tilde{r_o}(k)$ to the actual applied RM rate $r_o(k)$:

$$r_o(k) = \frac{(1-\tilde{r_o}(k))q_o^{\mathrm{R,min}}(k) + \tilde{r_o}(k)q_o^{\mathrm{R,max}}(k)}{Q_0}\,, \tag{3.19}$$

with the minimum on-ramp flow $q_o^{\mathrm{R,min}}(k)$ (veh/h) defined by the minimum allowed RM rate and the minimum required RM rate to prevent the on-ramp queue required to prevent the on-ramp queue from exceeding its maximum:

$$q_o^{\mathrm{R,min}}(k) = \max\left[r^{\min}Q_0, \frac{w_o(k) + d_o(k)T - w_o^{\max}}{T}\right]. \tag{3.20}$$

The maximum on-ramp flow $q_o^{\mathrm{R,max}}(k)$ (veh/h) is defined similarly as in (3.10):

$$q_o^{\mathrm{R,max}}(k) = \min\left[d_o(k) + \frac{w_o(k)}{T}, Q_0, Q_0\frac{\rho_m^{\max} - \rho_{m,1}(k)}{\rho_m^{\max} - \rho_m^{\mathrm{crit}}}\right] \tag{3.21}$$

3.2.5 Objective function and constraints

The objective of the controller is to minimize the Total Time Spent (TTS) by all the vehicles on the freeway by changing the VSLs and RM rates over the time steps $k^{\mathrm{c}} = k^{\mathrm{u}}C^{\mathrm{t}}+1, \ldots, k^{\mathrm{u}}C^{\mathrm{t}}+N^{\mathrm{p}}$. The following objective function $J(k^{\mathrm{u}})$ expresses the TTS:

$$J(k^{\mathrm{u}}) = T\sum_{\hat{k}=k^{\mathrm{u}}C^{\mathrm{u}}+1}^{k^{\mathrm{u}}C^{\mathrm{u}}+N^{\mathrm{p}}C^{\mathrm{c}}}\left\{\sum_{(m,i)\in I^{\mathrm{links}}}\rho_{m,i}(\hat{k})L_m\lambda_m + \sum_{o\in I^{\mathrm{orig}}}w_o(\hat{k})\right\}. \tag{3.22}$$

Here, the set I^{links} (-) is the set of indices of all pairs of segments and links, the set I^{orig} (-) is the set of all origin indices, and the set I^{ramps} is the set of all on-ramp indices.

Using this objective function the MPC optimization problem can be formulated:

$$\min_{\bar{u}(k^{\mathrm{u}})} J(k^{\mathrm{u}})$$

Subject to

Model: Eq. (3.3) – Eq. (3.21),

Initial states and disturbances:

$$\rho_{m,i}(k^{\mathrm{u}}C^{\mathrm{u}}),\ v_{m,i}^{\mathrm{ctrl}}(k^{\mathrm{u}}C^{\mathrm{u}}),\ \rho^{\mathrm{DS}}(\hat{k}),\ d_o(\hat{k}),$$

Constraints:

$$B^{\mathrm{L}} \le A\bar{u}(k^{\mathrm{u}}) \le B^{\mathrm{U}}. \tag{3.23}$$

The matrix A and vectors B^{L} and B^{U} represent the linear inequality constraints on the VSL and RM control signal respectively as detailed in the next subsections. The control signal $\bar{u}(k^{\mathrm{u}})$ is a vector consisting of the parameters of the head and tail of the speed-limited area and the parameters of the feedback control laws of the different on-ramps as will be detailed in the next subsections.

VSL signal and constraints

The evolution of the head and tail of the speed-limited area is described by the initial location of the head $x^{\mathrm{H,sl}}(k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}})$ (km) and tail $x^{\mathrm{T,sl}}(k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}})$ (km), and the speed $v^{\mathrm{H,sl}}(k^{\mathrm{c}})$ (km/h) and $v^{\mathrm{T,sl}}(k^{\mathrm{c}})$ (km/h) of the head and tail over time respectively. After the control horizon N^{c}, until the prediction horizon N^{p}, the speed of the head and tail locations are assumed to remain constant:

$$v^{\mathrm{H,sl}}(k^{\mathrm{c}}) = v^{\mathrm{H,sl}}(k^{\mathrm{u}}C^{\mathrm{t}}+N^{\mathrm{c}}) \text{ if } k^{\mathrm{c}} > k^{\mathrm{u}}C^{\mathrm{t}}+N^{\mathrm{c}}\,, \tag{3.24}$$

$$v^{\mathrm{T,sl}}(k^{\mathrm{c}}) = v^{\mathrm{T,sl}}(k^{\mathrm{u}}C^{\mathrm{t}}+N^{\mathrm{c}}) \text{ if } k^{\mathrm{c}} > k^{\mathrm{u}}C^{\mathrm{t}}+N^{\mathrm{c}}\,. \tag{3.25}$$

Based on the control vector, the location of the head and the tail of the control scheme at every time step k can be computed:

$$x^{\mathrm{H,sl}}(k) = x^{\mathrm{H,sl}}(k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}) + \sum_{j=k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}+1}^{k^{\mathrm{c}}} v^{\mathrm{H,sl}}(\lfloor (j-1)/C^{\mathrm{c}} \rfloor)T\,, \tag{3.26}$$

$$x^{\mathrm{T,sl}}(k) = x^{\mathrm{T,sl}}(k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}) + \sum_{j=k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}+1}^{k^{\mathrm{c}}} v^{\mathrm{T,sl}}(\lfloor (j-1)/C^{\mathrm{c}} \rfloor)T\,. \tag{3.27}$$

Several constraints have to be respected when optimizing the VSLs. First of all, the position of the head and tail have to lie within the upstream bounds $x^{\mathrm{H},0}$ (km) and $x^{\mathrm{T},0}$ (km) and downstream bounds $x^{\mathrm{H,end}}$ (km) and $x^{\mathrm{T,end}}$ (km):

$$x^{\mathrm{H},0} \le x^{\mathrm{H,sl}}(k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}) \le x^{\mathrm{H,end}}\,, \tag{3.28}$$

$$x^{\mathrm{T},0} \le x^{\mathrm{T,sl}}(k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}) \le x^{\mathrm{T,end}}\,. \tag{3.29}$$

If at time step $k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}$ the speed limits are not active or cover only 1 segment, i.e., when $x^{\mathrm{H,sl}}(k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}|k^{\mathrm{u}}-1)-1 \le x^{\mathrm{T,sl}}(k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}|k^{\mathrm{u}}-1)$, then these bounds are equal to the upstream x_0 (km) and downstream end of the freeway x_{end} (km). The notation $(\ldots|k^{\mathrm{u}}-1)$ indicates the control signal that is computed at time step $k^{\mathrm{u}}-1$. However, when the speed limits are active at control step $k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}$, then the location of the head $x^{\mathrm{H,sl}}(k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}|k^{\mathrm{u}})$ and tail $x^{\mathrm{T,sl}}(k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}|k^{\mathrm{u}})$ at control step $k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}$ should be equal to the previously computed values $x^{\mathrm{H,sl}}(k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}|k^{\mathrm{u}}-1)$ and $x^{\mathrm{T,sl}}(k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}|k^{\mathrm{u}}-1)$. In that case, the constraints are set to the following:

$$x^{\mathrm{H},0} = x^{\mathrm{H,sl}}(k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}|k^{\mathrm{u}}-1)\,, \tag{3.30}$$

$$x^{\mathrm{H,end}} = x^{\mathrm{H,sl}}(k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}|k^{\mathrm{u}}-1)\,, \tag{3.31}$$

$$x^{\mathrm{T},0} = x^{\mathrm{T,sl}}(k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}|k^{\mathrm{u}}-1)\,, \tag{3.32}$$

$$x^{\mathrm{T,end}} = x^{\mathrm{T,sl}}(k^{\mathrm{u}}C^{\mathrm{u}}+C^{\mathrm{c}}|k^{\mathrm{u}}-1)\,. \tag{3.33}$$

Secondly, the head and tail are allowed to propagate downstream with at most v^{eff} (km/h) or to propagate upstream with any speed so that they cannot 'overtake' a speed-limited vehicle:

$$v^{\mathrm{H,sl}}(k^{\mathrm{c}}) \le v^{\mathrm{eff}}\,, \tag{3.34}$$

$$v^{\mathrm{T,sl}}(k^{\mathrm{c}}) \le v^{\mathrm{eff}}\,. \tag{3.35}$$

Thirdly, the position of the head should be at or more downstream than the initial position of the tail:

$$x^{\mathrm{H,sl}}(k) \geq x^{\mathrm{T,sl}}(k)\,. \tag{3.36}$$

RM constraints

The RM control signal of an individual on-ramp consists of the switching times $t_{o,1}^{\mathrm{switch}}(k^{\mathrm{u}})$, $t_{o,2}^{\mathrm{switch}}(k^{\mathrm{u}})$, and $t_{o,3}^{\mathrm{switch}}(k^{\mathrm{u}})$, and the set-points $\rho_{o,1}^{\mathrm{set}}(k^{\mathrm{u}})$ and $\rho_{o,2}^{\mathrm{set}}(k^{\mathrm{u}})$.

By varying these parameters, the RM rate is affected. Several constraints on these parameters are included. First, it has to hold that the set-points $\rho_{o,i^{\mathrm{p}}}^{\mathrm{set}}(k^{\mathrm{u}})$ should be between 0 and the maximum set-point $\rho_{o,i^{\mathrm{p}}}^{\mathrm{set,max}}$ (veh/km/lane):

$$0 < \rho_{o,1}^{\mathrm{set}}(k^{\mathrm{u}}) \leq \rho_{o,1}^{\mathrm{set,max}} \tag{3.37}$$
$$0 < \rho_{o,2}^{\mathrm{set}}(k^{\mathrm{u}}) \leq \rho_{o,2}^{\mathrm{set,max}}\,. \tag{3.38}$$

Secondly, the switching time instants need to be constrained. Two cases are possible. The first case is that no RM is active at time step k^{c}. Then, it should hold that:

$$k^{\mathrm{u}}T^{\mathrm{u}} + T^{\mathrm{c}} \leq t_{o,1}^{\mathrm{switch}}(k^{\mathrm{u}}) \leq k^{\mathrm{u}}T^{\mathrm{u}} + N^{\mathrm{p}}T^{\mathrm{c}} \tag{3.39}$$
$$t_{o,1}^{\mathrm{switch}} + T^{\mathrm{c}} \leq t_{o,2}^{\mathrm{switch}}(k^{\mathrm{u}}) \leq k^{\mathrm{u}}T^{\mathrm{u}} + N^{\mathrm{p}}T^{\mathrm{c}} \tag{3.40}$$
$$t_{o,2}^{\mathrm{switch}} + T^{\mathrm{c}} \leq t_{o,3}^{\mathrm{switch}}(k^{\mathrm{u}}) \leq k^{\mathrm{u}}T^{\mathrm{u}} + N^{\mathrm{p}}T^{\mathrm{c}} \tag{3.41}$$

The second case is that RM is active at time step k^{u}. This is the case when $t^{\mathrm{ini}}(k^{\mathrm{u}}) = \max(t_{o,1}^{\mathrm{switch}}(k^{\mathrm{u}}-1), k^{\mathrm{u}}T^{\mathrm{u}}) < k^{\mathrm{u}}T^{\mathrm{u}} + T^{\mathrm{c}}$ and $t_{o,3}^{\mathrm{switch}}(k^{\mathrm{u}}-1) \geq k^{\mathrm{u}}T^{\mathrm{u}} + T^{\mathrm{c}}$. In this case the MPC should not be able to change $t_{o,1}^{\mathrm{switch}}(k^{\mathrm{u}})$ because it lies within the current time step $k^{\mathrm{u}}C^{\mathrm{t}}$ that cannot be affected. This is realized by the following constraints:

$$t^{\mathrm{ini}}(k^{\mathrm{u}}) \leq t_{o,1}^{\mathrm{switch}}(k^{\mathrm{u}}) + T^{\mathrm{c}} \leq t^{\mathrm{ini}}(k^{\mathrm{u}}) \tag{3.42}$$
$$k^{\mathrm{u}}T^{\mathrm{u}} + T^{\mathrm{c}} \leq t_{o,2}^{\mathrm{switch}}(k^{\mathrm{u}}) \leq k^{\mathrm{u}}T^{\mathrm{u}} + N^{\mathrm{p}}T^{\mathrm{c}} \tag{3.43}$$
$$t_{o,2}^{\mathrm{switch}} + T^{\mathrm{c}} \leq t_{o,3}^{\mathrm{switch}}(k^{\mathrm{u}}) \leq k^{\mathrm{u}}T^{\mathrm{u}} + N^{\mathrm{p}}T^{\mathrm{c}} \tag{3.44}$$

3.3 Simulation experiments

Simulations are carried out in order to investigate the controller behavior and performance in terms of CPU time used and TTS improvement of the controller. To this end, several simulations are performed in which the traffic situation and controller set-up is varied.

The main topic for investigation is the trade-off between the computation time and the realized throughput improvement. To this end, the parameterized MPC (PMPC) strategy is compared with a nominal MPC (NMPC) strategy that directly optimizes

the individual VSL values and RM rates. The NMPC strategy is expected to realize a similar or higher TTS when given sufficient CPU time. In order to obtain a fair comparison, both the control strategies are given the same CPU time budgets. It is expected that the PMPC strategy is able to obtain similar throughput improvement in less CPU time budget.

Additionally, the performance is compared when considering different controller set-ups, namely RM-only, VSL-only, and integrated RM and VSL, and when applied to different traffic situations, i.e., when resolving a jam wave – as done by the SPECIAL-IST algorithm – or by preventing congestion due to a high on-ramp flow. This allows to evaluate the added value of integrating the control measures in different scenarios. It is expected that integrated RM and VSL can realize the best throughput improvement because it has a larger control freedom, but that it does not necessarily lead to the best trade-off between computation time and realized throughput.

3.3.1 Simulation set-up

Figure 3.4 provides an overview of the simulation set-up. The extended METANET model as detailed in this paper is used as the process model – i.e., the real-world – and the prediction model. When implemented as the process model, three small changes are made. First of all, the parameter $\gamma_{m,i}(k)$ is set to 1 in the process model if $\gamma_{m,i}(k) > 0.1$ such that the entire segment is either speed-limited or not in order to reproduce the discrete spacing of the variable message signs. Secondly, the switching times are rounded to the nearest multiple of T that is less than or equal to the switching time. Thirdly, a lead-in procedure is introduced for the VSLs preventing too large speed drops on the freeway. To this end, the VSL value of a gantry is set to the minimum of the desired VSL and the VSL value of the downstream gantry increased with 10 km/h which is iteratively computed from downstream to upstream.

A 20 km long freeway with 2 on-ramps and 2 off-ramps is considered as shown in Figure 3.5. The freeway consists of three origins and 20 identical segments with a length of 1 km and 2 lanes. Every segment has the same parameters, adopted from [Kotsialos et al., 2002a], namely: $T = 10$s, $\tau = 18$ s, $\kappa = 40$ (veh/km/lane), $\rho^{\text{crit}} = 33.5$ veh/km/lane, $a_m = 1.867$, $v^{\text{free}} = 102$ km/h, $\eta^{\text{high}} = 65$ km/h^2, $\eta^{\text{low}} = 30$ km/h^2. Using these parameters, a capacity of 2000 veh/h/lane is realized and a capacity drop can be observed. The freeway traffic is simulated for scenarios of 3 hours. All the segments can be controlled by means of VSLs. The value of the effective speed limit v^{eff} is set to 50 km/h. The two on-ramps are controlled by means of RM. The minimum RM rate is set to 0.05, the feedback gains of the PMPC strategy are set to $K^{\text{s}}_{o,i^{\text{p}}} = 0.5$, and the maximum density set-point is set to $\rho^{\text{set,max}}_{o,i^{\text{p}}} = 60$ (veh/km/lane).

The process and prediction model sampling time steps T are set to 10 seconds. The control signal update time step T^{u} is set to 300 seconds, and the control time step T^{c} is set to 60 seconds. This means that every 300 seconds the control signal is optimized

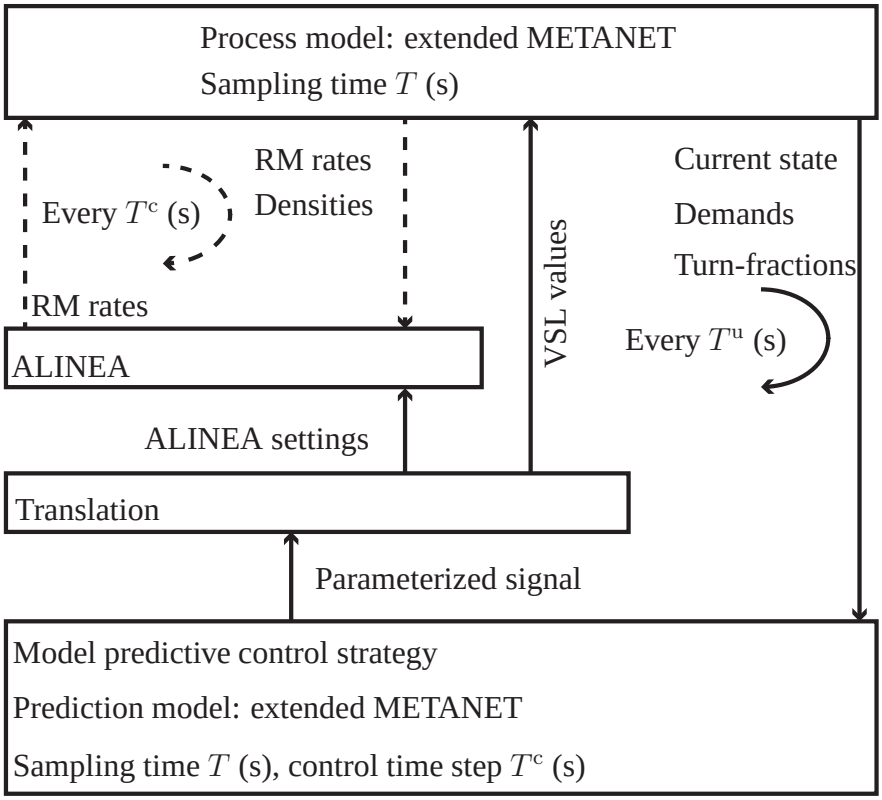


Figure 3.4: Simulation set-up

Figure 3.5: Simulation network

based on the current traffic state. The values of the control signals are allowed to change every 60 seconds.

It is assumed that no measurement noise affects the traffic state used by the MPC strategy. Also, it is assumed that a prediction of the demand and turn-fractions is available for the MPC strategy.

The evaluation is carried out using Matlab R2015a on a computer with a 3.6 GHz processor, 8 cores, and 16 Gb RAM. For the optimization the Sequential Quadratic Programming algorithm of the fmincon solver of the MATLAB optimization toolbox is used, the function tolerance is set to $5 \cdot 10^{-3}$ and the step tolerance is set to $1 \cdot 10^{-7}$. Parallel computing on 8 cores is used to determine the numerical derivative of the objective function. In order to realize a fair comparison, both approaches are given the same amount of CPU time in which they can find the optimal solution. When this computation time is not reached, the optimization is repeated from a new, randomly selected starting point. When the computation time is exceeded, the optimization is stopped. The best solution out of the different starts is applied to the process. All the simulations are repeated with three budgets, namely 300, 600, 1200, 1800, and 3600 seconds. To speed up the simulations, parallel computing is used to compute the gradient. For a fair comparison, the CPU time budget reflects the total computation time used by all the cores. Because the computations are carried out in parallel, the actual elapsed time is smaller than the CPU time budget.

The NMPC approach is implemented as follows. Similar as in [Kotsialos et al., 2005], the RM rate $\tilde{r_o}(k)$ (-) of an on-ramp is directly optimized. It is bounded between 0 and 1 and constrained in such a way that the RM can change with a maximum of 0.25 per control step. The optimized RM rate $\tilde{r_o}(k)$ (-) is translated to the actual applied RM rate $r_o(k)$ using (3.19). The VSL strategy proposed in [Hegyi et al., 2005a] is implemented. The VSL values are bounded so that they are larger than 50 km/h and smaller than the free flow speed. Additionally, the following constraint is included $v_{m,i}^{\mathrm{ctrl}}(k^{\mathrm{c}}) \leq v_{m,i+1}^{\mathrm{ctrl}}(k^{\mathrm{c}})+10$ preventing sudden speed drops in the downstream direction.

3.3.2 Case I: jam wave

A scenario in which a jam wave is present on the freeway is evaluated. Figure 3.7 (a)–(f) shows the no-control situation in which a jam wave enters the freeway at the most downstream end. This jam is created by increasing the density at the downstream end of the freeway from time 380 s to 1080 s. The demand at the origins is equal to 3800 veh/h, 455 veh/h, and 400 veh/h until time 5500 s for the mainstream origin, on-ramp 1 (O1) and on-ramp 2 (O2) respectively. The percentage of traffic exiting at the off-ramps is 10% and 12% for off-ramp 1 and 2 respectively. After time 5500 s the demands decrease to 3500 veh/h, 240 veh/h, and 260 veh/h respectively. The capacity drop due to this jam wave, determined using simulation experiments, is approximately 5.6%. The total time spent of the no-control scenario is 3325.1 veh·h.

Various control set-ups are tested in the control scenario. In order to evaluate the performance and behavior of the MPC strategy when resolving a fully formed jam wave, so that we can interpret the solution, which is expected to be similar to the solution of SPECIALIST, the controller is started after 1500 seconds. Note that this represents an artificial situation, since in practice the MPC strategy is always active so that it will start controlling before the jam wave is fully formed. The maximum on-ramp queue length is set to 150 vehicles for both ramps. The prediction horizon is set to 4800 seconds and the control horizon is set to 2400 seconds. The control horizon is not applicable to the parameterized RM strategy, because a specific choice of the switching times fully determines the controller behavior over any horizon. Note that the VSL control signal is allowed to change every 60 seconds so that 40 steps are to be optimized.

The control horizon is not relevant for the parameterized RM strategy, since it optimizes the switching time instants when the feedback RM strategy is changed instead of explicitly optimizing the RM rates at the control sampling time steps.

Table 3.1 presents the quantitative results for the different computation time budgets. It can be observed that a computation time budget of 1200 seconds is sufficient for all the parameterized strategies to realize the best throughput improvement. The average elapsed times per controller update for these budgets are all below 300 seconds. The RM-only NMPC strategy achieves similar performance as the PMPC strategy for a budget of 3600 seconds. However, even the budget of 3600 seconds does not seem to be enough for VSL-only or integrated VSL and RM using the NMPC strategy.

The qualitative results of the VSL-only case shown in Figures 3.7 (g)–(l) show that the jam wave is resolved by imposing a speed-limited area upstream of the jam wave, similar as done by the SPECIALIST algorithm. Figures 3.7 (m)–(r) show the results of resolving the jam wave using the RM-only strategy. It can be seen that it takes longer for the RM-only to resolve the jam wave explaining the lower TTS gain. Figures 3.7 (s)–(x) show that the integration of VSLs and RM reduces the application of VSLs upstream of on-ramp 1, as well as the time over which VSLs are needed. Figure 3.6 (a) shows the total network outflows for the different control strategies. It can be seen that it takes longer for the RM-only strategy to improve the total network outflow. Also, it can be seen that the integration of VSL and RM reduces the initial outflow reduction and a quicker outflow increase after the jam has resolved when compared to the VSL-only case, explaining the TTS improvement.

3.3.3 Case II: bottleneck

The second case consists of a traffic jam caused by a too high on-ramp flow. The no control situation is shown in Figure 3.8 (a)–(f). The origin demands were set to 3900 veh/h, 455 veh/h, and 390 veh/h for the mainstream origin, on-ramp 1 (O1), and on-ramp 2 (O2) respectively, and they were taken to be constant until time 4500 s except

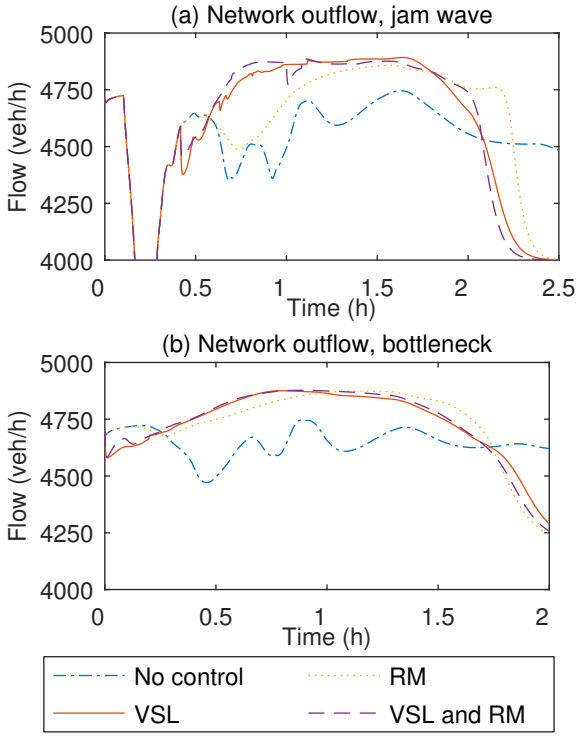


Figure 3.6: Network outflow in (a) the jam wave and (b) the bottleneck case using different control strategies.

for on-ramp 1 of which the inflow from time 1500 s to 2000 s was increased to 1500 veh/h triggering a traffic jam. The percentage of traffic exiting at the off-ramps is 10% and 12% for off-ramp 1 and 2 respectively. After time 4500 s the demands decreased to 3500 veh/h for the mainstream origin and to 260 veh/h for on-ramp 2. The resulting TTS is 2536.0 veh·h.

Several control set-ups are evaluated in the control situation. The maximum on-ramp queue lengths were set to 75 and 20 vehicles for on-ramp 1 and 2 respectively. Due to this, coordination between the two on-ramps is required, since the capacity of on-ramp 2 is not sufficient to prevent congestion on its own. The prediction horizon is set to 4800 seconds and the control horizon is set to 2400 seconds.

Table 3.1 presents the quantitative results for the different computation time budgets. It can be observed that for VSL-only and integrated VSL and RM set-ups the PMPC realizes higher TTS gains in shorter budgets. For the RM-only case, both the NMPC and PMPC realize similar TTS improvements, namely 9.9% and 9.7% respectively for short CPU time budget of 300 seconds.

The qualitative results of the VSL-only strategy shown in Figures 3.8 (g)–(l) indicate the ability to prevent bottleneck congestion by imposing a speed-limited area upstream of on-ramp 2. Figures 3.8 (m)–(r) show that in the RM-only case on-ramp 1 starts metering immediately so that this flow reduction arrives at on-ramp 2 when the demand increases. The qualitative results of the integrated VSL and RM case in Figures 3.8 (s)–(x) indicate that the integration of VSL and RM reduces the extent to which VSLs are

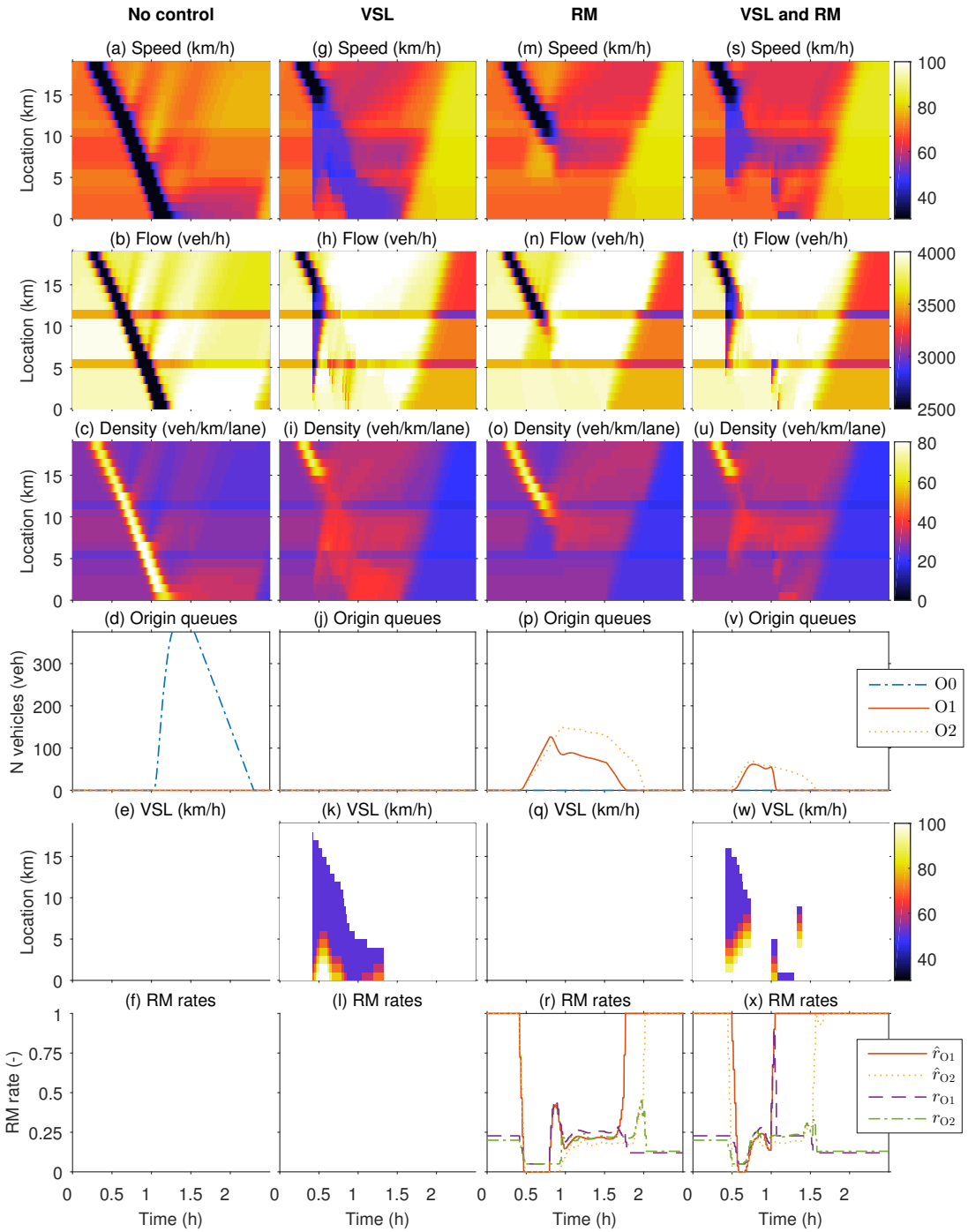


Figure 3.7: Results of the jam wave case using a CPU time budget of 3600 s. Every column represent a different control strategy. The first three rows show the contour plots of the traffic dynamics, the fourth row shows the origin queues, and the bottom 2 rows show the control signals.

imposed upstream of on-ramp 1. When comparing the outflow plots in Figure 3.6 (b) it can be seen that the integrated VSL and RM strategy limits the initial flow reduction when compared to the VSL-only strategy. It also shows that integrated VSL and RM is able to quicker restore the network outflow when compared to RM-only.

Table 3.1: Overview of quantitative results for both cases. The no control TTS is 3325.1 veh·h for the jam wave case and 2536.0 veh·h for the bottleneck case. The percentage gain in TTS for the closed-loop simulation compared to the no control situation and the average elapsed time (ET) per iteration are presented.

			CPU budget: 300 s		CPU budget: 600 s		CPU budget: 1200 s		CPU budget: 1800 s		CPU budget: 3600 s	
			% gain	ET (s)	% gain	ET (s)	% gain	ET (s)	% gain	ET (s)	% gain	ET (s)
Jam wave	PMPC	VSL	3.9	57.7	11.1	103.3	11.1	207.8	11.1	311.7	11.2	621.5
		RM	7.2	74.6	7.3	148.8	7.3	297.9	7.3	446.6	7.3	890.4
		VSL RM	10.8	57.6	11.2	110.1	11.6	209.5	11.7	316.7	11.9	624.6
	NMPC	VSL	1.0	54.4	5.5	127.7	8.7	181.0	9.5	291.9	10.1	521.1
		RM	4.1	48.0	4.5	94.7	5.2	179.7	5.2	267.3	7.3	528.3
		VSL RM	0.8	73.4	8.1	173.1	8.1	173.0	9.8	294.5	11.2	554.8
Bottleneck	PMPC	VSL	4.7	45.9	5.0	97.3	9.3	194.7	9.2	299.4	9.2	604.8
		RM	9.7	74.2	9.7	147.9	9.7	295.5	9.7	443.2	9.7	883.0
		VSL RM	9.1	46.3	9.6	88.6	9.8	195.1	10.0	293.3	10.0	605.5
	NMPC	VSL	-3.1	54.3	3.8	125.7	5.4	191.9	6.8	285.9	9.2	531.5
		RM	9.9	48.6	9.9	91.2	9.9	171.3	9.9	260.2	9.9	508.2
		VSL RM	-3.6	73.6	2.7	172.3	2.7	180.9	5.3	316.6	6.8	550.3

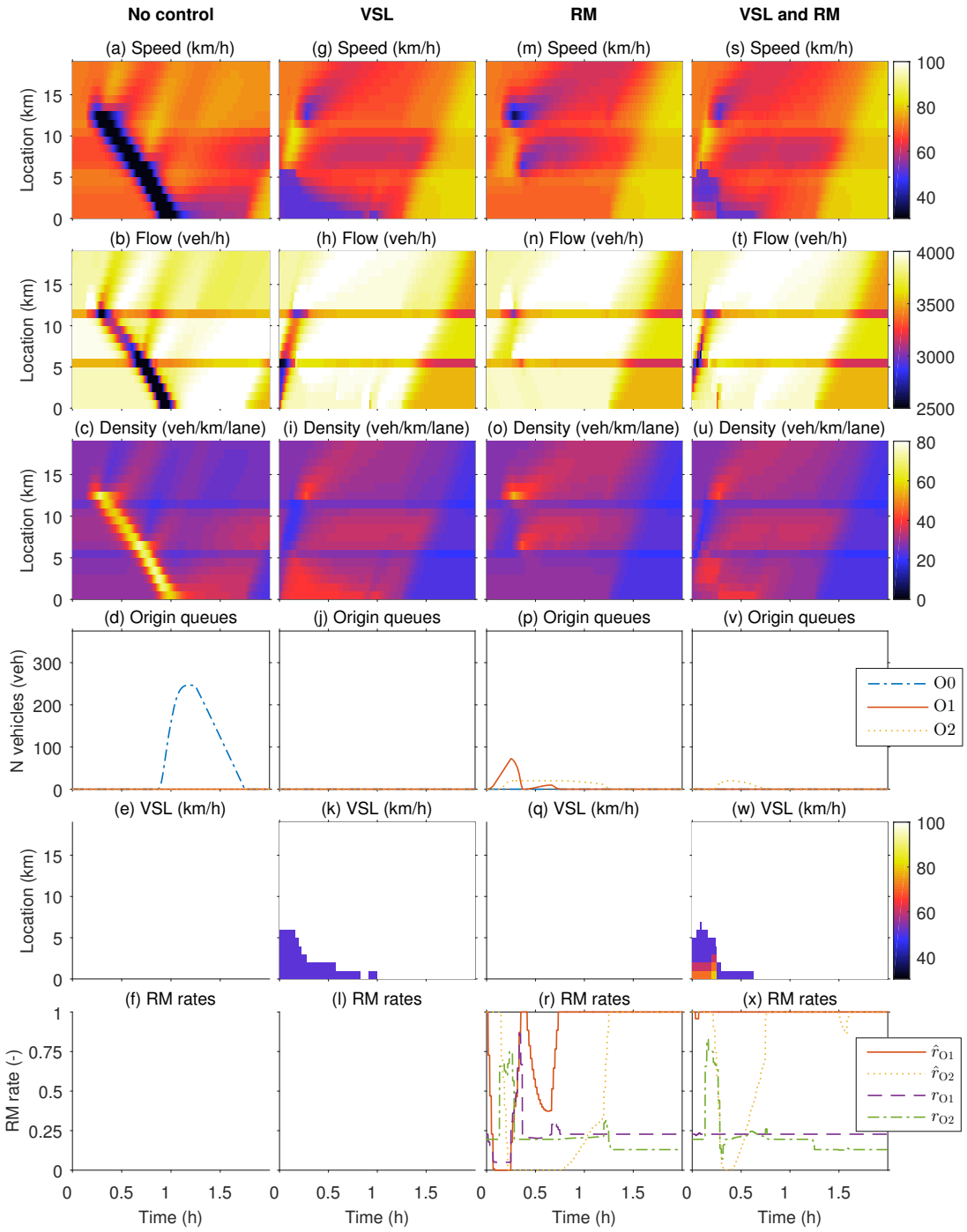


Figure 3.8: Results of the bottleneck case using a CPU time budget of 3600 s. Every column represent a different control strategy. The first three rows show the contour plots of the traffic dynamics, the fourth row shows the origin queues, and the bottom 2 rows show the control signals.

3.4 Conclusions and recommendations

In this paper the computation time of an MPC strategy for integrated RM and VSLs was improved considerably by parameterizing a control scheme based on ALINEA ramp metering and a SPECIALIST-like VSL control scheme. Due to this, the dimension

of the optimization problem has become independent of the number of VSL signs. Additionally, the number of parameters needed per on-ramp has become independent of the prediction horizon. Simulations have shown that the control approach proposed in this paper can achieve a better performance than a non-parameterized MPC strategy when using the same budget of computation time for VSL-only and integrated VSL and RM strategies. It was found that the non-parameterized strategy realizes a slightly better throughput improvement for the RM-only case.

In further research, the impact of noise and uncertainties on controller performance can be studied. When needed, a robust control design may have to be designed. Additionally, it can be studied how the approach can be extended to include multiple VSL areas when applying it to larger freeway networks. It is also recommended to compare the proposed strategy to simpler, uncoordinated or non-predictive strategies. Also, the use of in-vehicle technologies may lead to improved detection and actuation possibilities and potentially a reformulation of the control strategy. Future research can also investigate approaches to further improve the computation time, for instance, using a problem-tailored algorithm to solve the optimization problem as discussed in [Kotsialos et al., 2002b].

Acknowledgment

This work is part of the research programme ‘The Application of Operations Research in Urban Transport’, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

Part II

Urban traffic control

Chapter 4

Linear MPC-based Urban Traffic Control using the Link Transmission Model

In this chapter an optimization-based control strategy is developed for the optimization of flows in order to improve the urban network throughput. The developed strategy is used as a basis for the control strategies proposed in the next two chapters. This chapter is based on the following paper that is currently under review:

G.S. van de Weg, M. Keyvan-Ekbatani, A. Hegyi, and S.P. Hoogendoorn, Linear MPC-based Urban Traffic Control using the Link Transmission Model. *Transactions on Intelligent Transportation Systems*, submitted 2017-6-12.

Abstract

In this paper we develop a computationally efficient model predictive control (MPC) strategy for optimization of intersection flows to improve the urban traffic network throughput. Several linear and quadratic MPC approaches have been developed in the literature to reduce the computational complexity of the problem, but without considering the back propagating waves caused by spillback. Thus, the principal contribution of this paper is the formulation of a linear optimization problem for an MPC strategy that considers downstream propagating waves in free flow traffic, queuing dynamics, and upstream propagating waves caused by spillback. The linear optimization problem is obtained by describing link dynamics using the link transmission model, and aggregating the traffic dynamics to (several) tens of seconds. The performance of the proposed controller is compared with two other existing strategies; a store-and-forward

model-based and a cell transmission model (CTM)-based approach. The total time spent (TTS) by all the vehicles in the network and the computation time have been applied as performance indexes for the appraisal of the control strategies. Simulation results show that including upstream propagating waves results in better controller performance due to inclusion of the impact of link outflow on maximum link inflow. It is also shown that the control approach realizes a higher throughput while using less computation time compared to a CTM-based approach. The comparison with a store-and-forward model-based optimization approach revealed that the proposed strategy can realize higher throughput but may require more computation time.

4.1 Introduction

The performance – e.g. throughput, pollution, safety, reliability – of urban road traffic networks is in many occasions not optimal. This paper focuses on improving the performance of urban road traffic networks using urban traffic control (UTC) for coordinating the intersection interaction. A common example of coordination is the creation of green waves in order to reduce the delay of high-volume traffic flows which is mainly effective in undersaturated traffic conditions [Little, 1966].

This paper proposes a control algorithm that is able to:

1. achieve good network performance in various traffic regimes, such as, undersaturated, saturated, and oversaturated traffic. More specifically, it should correctly handle forward moving waves and backward moving waves, such as queue spillback and gridlock
2. it should have sufficiently low computation time to be applied in real-time for larger networks.

4.1.1 Overview of urban traffic control strategies

One of the complicating factors of UTC is that intersections influence each other differently in various traffic regimes. Similar to the definitions in [Aboudolas et al., 2010], this paper categorizes the traffic states in the links as follows: undersaturated, saturated, and oversaturated. Note that the definition used here refers to a single link while in [Aboudolas et al., 2010] a regime refers to the traffic condition of the majority of the links in the network.

The *undersaturated* regime represents the situation in which a queue can be emptied during a green time implying that a coupling from upstream to downstream intersections exists. This is exploited by strategies that create green waves, such as MAXBAND [Little, 1966]. Other widely used strategies that are mainly effective in undersaturated regimes and applicable to large-scale networks are SCOOT and SCATS

[Hunt et al., 1982, Luk, 1984]. According to Papageorgiou et al. [2003] the performance of SCOOT deteriorates in the saturated traffic regime.

The *saturated* regime is defined as the situation in which queues cannot be dissolved during a green time implying that no direct coupling between intersections exists. The recently proposed max-pressure (or back-pressure) algorithms use this mechanism to distribute queues in an urban network [Varaiya, 2013, Le et al., 2015]. An advantage of that strategy is that it is distributed and requires only data gathered in the vicinity of the intersection. Gregoire et al. [2014] extended the max-pressure algorithm to deal with oversaturated regimes as well. The TUC strategy is a noteworthy example of a practice implemented control strategy designed for saturated regimes that is also capable of creating green waves [Diakaki et al., 2003]. Later, Aboudolas et al. [2010] formulated the problem of network-wide signal control as a quadratic programming problem that aims at minimizing and balancing the link queues so as to minimize the risk of queue spillback.

The *oversaturated* regime is characterized by queues which propagate to upstream intersections causing a coupling from downstream intersections to upstream intersections. This coupling is time delayed due to the accelerating behavior of vehicles which is typically described by upstream propagating waves. Due to this time delay, the actual number of vehicles that can be stored in a congested link is typically smaller when compared to the maximum storage capacity. This coupling can result in congestion propagation through a larger part of the network or even gridlock [Daganzo, 2007]. Gayah et al. [2014] showed that in an extremely congested network, adaptive traffic signals might have little to no effect on the network performance due to downstream congestion and queue spillback. Hence, other strategies such as gating or perimeter control might be beneficial to alleviate instability under oversaturated traffic regimes. Recently, Geroliminis and Daganzo [2008] found evidence for the existence of a network fundamental diagram (NFD) for urban traffic networks which has been exploited as a basis for the derivation of urban signal control approaches for oversaturated regimes. The combination of the NFD concept with gating or perimeter control of traffic flow lead to control strategies that deal with oversaturated regimes (see [Keyvan-Ekbatani et al., 2012] for single region gating; [Geroliminis et al., 2013, Hajiahmadi et al., 2015a] for multi-region, [Keyvan-Ekbatani et al., 2015b] for multiple concentric regions, and [Keyvan-Ekbatani et al., 2015a] for remote perimeter control with large control steps).

The aforementioned control strategies differ not only in the extent to which they have been tested in the field and the underlying algorithmic formulations, but also in the exploited control mechanism. A potential challenge of many of these strategies is that they are mainly effective in only one or two of the traffic regimes.

4.1.2 Overview of model-based optimal control strategies

Traffic control strategies that aim at improving the urban network performance in all traffic regimes can benefit from predicting the impact of a control strategy over a time horizon. The reason for this is that a change in the outflow of one intersection can affect the outflow of another upstream or downstream intersection in the future. Model-based optimal control techniques are especially suited to take these effects into account.

Model predictive control is a popular type of model-based optimization technique [Garcia et al., 1989, Mayne et al., 2000]. At every controller sampling time step, the optimal control signal is obtained and this signal is applied to the process. When new measurements become available this process is repeated, this is called the receding horizon principle. Some advantages of MPC are that it has a feedback structure, different types of objective functions can be specified, it explicitly predicts the process dynamics, and it is relatively easy to include constraints. However, several challenges exists as well, such as, the computational complexity of the optimization problem, the model-reality mismatch – i.e., the mismatch between predicted states and realized states –, and the uncertainty in predicting the disturbances. See [Burger et al., 2013] for an overview of considerations for applying MPC to traffic control. The design of MPC strategies which are able to improve the performance of processes that are subject to noise and uncertainties is commonly referred to as robust MPC (see [Bemporad and Morari, 1999] for a survey of the robust MPC literature).

To apply MPC for urban road networks various approaches have been proposed in the literature. These approaches use as an input the current traffic state and require a prediction of the demand and of the turn fractions or routes of the traffic. One of the main differences between these approaches are the models that are used for the prediction of the traffic states and the features – such as, the macroscopic traffic flow characteristics – that are considered in these models. In many cases it holds that adding more features (that improve the match with the reality) leads to better controller performance, because the model-reality mismatch is reduced. The application of a model with more features may lead to higher computational complexity, depending on the structure of the resulting optimization problem. Thus, finding a good balance between controller performance and computational complexity is an important challenge when developing MPC strategies for urban traffic control.

The MPC strategies of Lo [1999] and Van den Berg et al. [2007] consider signal timings as decision variables. Lo [1999] formulated a mixed-integer linear programming (MILP) problem based on the Cell-Transmission Model to optimize the signal timings. Van den Berg et al. [2007] proposed a non-linear MPC based on a detailed traffic flow model – which is an extension of the model of Kashani and Saridis [1983] – in order to optimize the network throughput in all regimes. These approaches require high computation times because of the detailed traffic model that is used.

Lin et al. [2012], Le et al. [2013], and Aboudolas et al. [2010] addressed this problem by assuming that the turn fractions – i.e., the distribution of link outflow to direct

downstream links – are known. Also, they aggregated the traffic flow dynamics to several (tens of) seconds and used green-splits as control signals instead of considering binary control signals indicating whether a link has a green light (1) or a red light (0). Lin et al. [2012] proposed a non-linear MPC strategy based on a simplified version of the model of Van den Berg et al. [2007], called the S-Model. This non-linear MPC approach was cast as a MILP problem in [Lin et al., 2011] which can be more efficiently solved. Le et al. [2013] proposed a linear-quadratic MPC strategy for the optimization of both signal settings and turn fractions in all traffic regimes. Aboudolas et al. [2010] also proposed a linear-quadratic MPC strategy based on the store-and-forward model for traffic flow optimization in saturated regimes which can be efficiently solved.

The aforementioned approaches employ different traffic models with different features resulting in different trade-offs in controller performance and computation time. One observation that can be made is that only the approach of Lo [1999] includes the impact of upstream propagating waves on the maximum link inflow by exploiting the CTM. The consequence of not including upstream propagating waves caused by spill back is that the maximum link inflow is overestimated which can be expected to cause a waste of green time in (over-)saturated regimes. This may affect the performance and efficiency of the optimization-based controllers.

4.1.3 Research objective and contributions

The aim of this research is designing a computationally efficient MPC strategy to control traffic flow under all traffic regimes which considers the impact of upstream propagating waves on the maximum link inflow. The control strategy is developed for medium to large-scale networks covering several tens of intersections. To this end, Section 4.2 shows that taking the upstream propagating wave speed, and the free flow travel time into account leads to a linear optimization problem with linear inequality constraints when assuming aggregated traffic dynamics and known turn fractions. This is realized by describing the link dynamics using the link transmission model (LTM) of Yperman [2007] and describing aggregated traffic dynamics. Hajiahmadi et al. [2015b] showed that an MPC strategy based on the LTM for freeway networks can be solved using a mixed-integer linear programming problem. The contribution of this paper is the formulation of a linear optimization problem for the control of link outflows in an urban road traffic network for the optimization of urban network throughput in all traffic regimes and evaluating the controller performance in terms of throughput improvements and computation time used. The approach is called LML-U which is an abbreviation for “Linear MPC using the LTM for Urban traffic control”.

In more detail, the main contributions of this paper are:

1. Design of a linear MPC strategy using the LTM for the optimization of aggregated traffic dynamics that considers downstream propagating waves caused by free flow dynamics, queuing dynamics, and upstream propagating waves caused

by spill back, see Section 4.2. The advantage of using the LTM to describe the traffic dynamics is that there is no need to divide a link into segments which is more efficient from a computational point of view compared to other approaches, e.g. based on the cell-transmission model (CTM).

2. Showing that the inclusion of upstream propagating waves can lead to better throughput improvements when compared to the approaches of Le et al. [2013] and Aboudolas et al. [2010] in Section 4.3.

3. Comparing the controller performance – in terms of computation time used and realized throughput improvements – to the control approaches of Le et al. [2013] and Aboudolas et al. [2010] in Section 4.3.

4. Showing that the approach can be applied to a large network in Section 4.3.

4.2 Model predictive control strategy design and formulation

The MPC strategy developed in this paper uses the LTM of Yperman [2007] as the prediction model. The LTM is chosen since it is capable modeling queuing dynamics and downstream and upstream propagating waves. Also, the LTM can be used to formulate an efficient optimization problem for two reasons as will be shown in this section, namely; 1) it can be used to formulate a linear optimization problem, and 2) it can describe the link dynamics using only two states.

The remainder of this section is structured as follows. First, Section 4.2.1 introduces the main assumptions. Then, Section 4.2.2 introduces the traffic flow modeling used to formulate the optimization problem. Section 4.2.3 formulates the linear optimization problem based on this model and Section 4.2.4 specifies the dimension of the linear optimization problem.

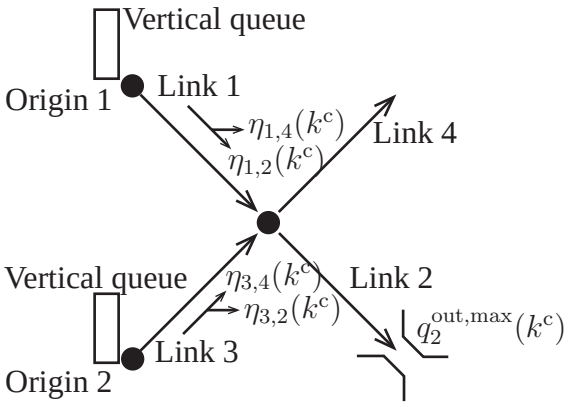


Figure 4.1: Example of the network elements.

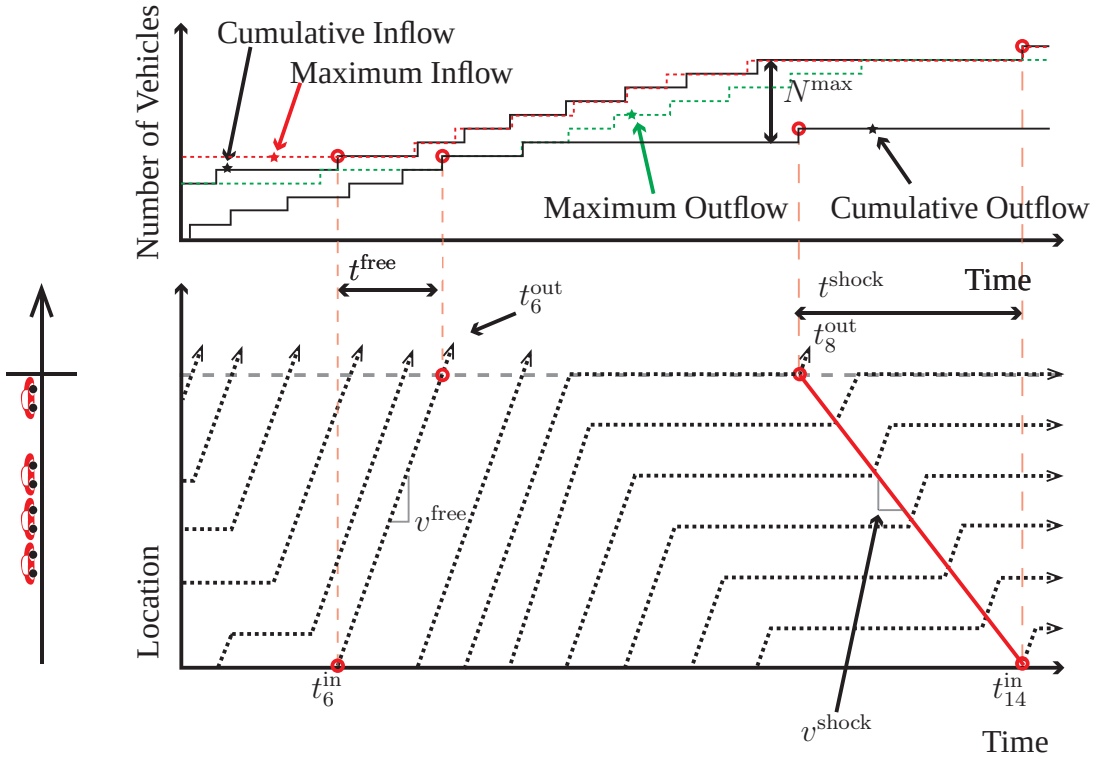


Figure 4.2: Time-space diagram (bottom) and plot of the cumulative curves (top) illustrating the most important variables used in this article. The constraints on the cumulative inflow and outflow curves – i.e., 'Maximum Inflow' and 'Maximum Outflow' – are also represented in this figure.

4.2.1 Assumptions

The following assumptions are made in this paper:

1. Aggregated traffic dynamics are considered by increasing the sampling time to several (tens of) seconds (in line with the assumptions in [Aboudolas et al., 2010, Lin et al., 2012, Le et al., 2013]). In this way, the number of time steps for predicting the traffic dynamics is reduced and no signal timings have to be considered which reduces the computational burden of the model. Nevertheless, the controller is able to account for upstream and downstream propagating waves and the distribution of queues over the networks when optimizing the intersection flows.

2. It is assumed that the turn fractions – i.e., the distribution of traffic from one link to its downstream links – are known and static – in other words, they are not influenced by the control signals.

3. In order to model the link dynamics, it is assumed that the saturation flow rates, the average free flow speeds, the upstream propagating wave speeds, and maximum link densities are known.

4. It is assumed that the demand is known and no disturbances or sources of uncertainty are present.

A discrete-time model is used in this paper. To this end, the time step k (-) refers to the period $t \in \big[Tk, T(k+1)\big)$ (h) where the time T (h) is the process model sampling time. The superscript $^{\mathrm{c}}$ refers to the timing of the prediction model used in the MPC strategy giving, e.g. k^{c} and T^{c}. The time steps T and T^{c} are related by the factor $\epsilon^{\mathrm{c}} = T^{\mathrm{c}}/T \in \mathbb{I}^{+}$. The time step T^{c} is in the range of 1 to 10 seconds and is limited by the link length due to the CFL conditions that have to be satisfied as explained in Section 4.2.2. The control signal is optimized at time steps k^{m} (-) with $k^{\mathrm{m}} = (k-1)\epsilon^{\mathrm{m}} + 1$ where the factor $\epsilon^{\mathrm{m}} = T^{\mathrm{m}}/T \in \mathbb{I}^{+}$ relates the controller sampling time T^{m} (h) to the process model sampling time T. It also holds that the factor $\epsilon^{\mathrm{c,m}} = T^{\mathrm{m}}/T^{\mathrm{c}} \in \mathbb{I}^{+}$ (-) relating the controller sampling time and the prediction model sampling time is a positive integer.

4.2.2 Traffic flow dynamics

This section details the description of the traffic flow dynamics. The main elements of the model used in this paper are links, origins, and nodes as illustrated in Figure 4.1. The approach also includes the possibility to impose restrictions on the outflow at exits of the network, see e.g. link 2 in Figure 4.1. The description of the link dynamics follows the LTM of Yperman [2007] which is briefly introduced in Section 4.2.2. In contrast to that model a node model to connect the links is not explicitly included, since, the connection between links is an outcome of the optimization problem. Section 4.2.2 details the formulation of the LTM using linear state equations and constraints.

Brief introduction to the LTM

The LTM describes the link dynamics using two traffic states, namely, the cumulative inflow $N_i^{\mathrm{L,in}}(k^{\mathrm{c}})$ (veh) and outflow $N_i^{\mathrm{L,out}}(k^{\mathrm{c}})$ (veh) of every link i (-) in the network. This is an advantage when compared to approaches that divide a link into segments which require much more traffic states to describe the link dynamics. More important may actually be that the numerical stability of these segment-based schemes requires a small time step due to the CFL condition. Since some segments are small, the simulation often needs to run with a small time step. Another advantage of the LTM is that it is capable of modelling all traffic regimes and specifically considers downstream and upstream propagating waves.

Figure 4.2 illustrates the description of the traffic dynamics in the LTM. It is assumed that the free-flow speed v_i^{free} (km/h) is known and constant, and that a vehicle cannot exit the link before the time t_i^{free} (h) that it requires to travel through the link with the free-flow speed as illustrated by the trajectory of vehicle 6. Thus, the maximum link outflow depends on the link inflow in the past. In saturated regimes, the link outflow is equal to the saturation rate q_i^{sat} (veh/h). Finally, in oversaturated regimes the wave

speed v_i^{shock} (km/h) is included as illustrated by the wave that starts when vehicle 8 exits the link. Note that it takes a time t_i^{shock} (h) for the upstream propagating wave caused by spill back to travel through the link. Due to this, vehicle 14 can only enter the link a time t_i^{shock} (h) after vehicle 8 has exited the link. The implies that the maximum link inflow depends on the outflow of the link in the past.

Thus, in order to model the traffic dynamics using the LTM it is required to know the cumulative inflow and outflow in the past. For instance, free-flow dynamics can be modeled by assuring that the cumulative outflow at time t is not larger than the cumulative inflow at time $t - t_i^{\text{free}}$ in the past as illustrated in Figure 4.2. The next subsection will formally describe the traffic flow modelling.

Formulation of the LTM using linear state equations and constraints

The above mentioned dynamics are modelled using linear state update equations of the cumulative curves and linear constraints. In order to realize this, the control variables used are the effective fractions of green time $b_i^{\text{L,eff}}(k^{\text{c}})$ (-) used by the links, and the effective fractions of green time $b_j^{\text{O,eff}}(k^{\text{c}})$ (-) used by the origin queues. For a link, this is defined as the realized link outflow $q_i^{\text{realized}}(k^{\text{c}})$ (veh/h) divided by the link saturation flow:

$$b_i^{\text{L,eff}}(k^{\text{c}}) = \frac{q_i^{\text{realized}}(k^{\text{c}})}{q_i^{\text{sat}}}\,. \tag{4.1}$$

By using the effective fractions it is possible to model the link dynamics using linear equations and include free flow travel times and upstream and downstream propagating waves by adding linear inequality constraints as detailed below. The optimization will take care of matching the outflows and inflows of links that are connected to each other so that the optimization problem is essentially serving as the node model.

The cumulative flow out of link i is updated as follows:

$$N_i^{\text{L,out}}(k^{\text{c}}+1) = N_i^{\text{L,out}}(k^{\text{c}}) + b_i^{\text{L,eff}}(k^{\text{c}}) q_i^{\text{sat}} T^{\text{c}}\,. \tag{4.2}$$

Note that this equation assumes that the link outflow is equal to the saturation rate. However, the effective fraction of green time $b_i^{\text{L,eff}}(k^{\text{c}})$ used enables to limit the outflow when there is no queue. In this way, free-flow dynamics can be modelled using the following constraint:

$$N_i^{\text{L,out}}(k^{\text{c}}+1) \leq \gamma_i^{\text{c,free}} N_i^{\text{L,in}}(k^{\text{c}} - k_i^{\text{c,free}} + 2) + (1-\gamma_i^{\text{c,free}}) N_i^{\text{L,in}}(k^{\text{c}} - k_i^{\text{c,free}} + 1)\,, \tag{4.3}$$

where the number of time steps $k_i^{\text{c,free}} = \lceil t_i^{\text{free}}/T^{\text{c}} \rceil$ (-), and the fraction $\gamma_i^{\text{c,free}} = k_i^{\text{c,free}} - t_i^{\text{free}}/T^{\text{c}}$ (-) the residual of a sampling time step that the free-flow travel time t_i^{free} (h) is exceeded by $k_i^{\text{c,free}}$. The mathematical operator $\lceil \cdot \rceil$ rounds the argument of the function to the nearest integer that it higher than the argument of the function. The

interpretation of this constraint is that the cumulative outflow curve should always lie below the cumulative inflow curve shifted with the free-flow travel time as illustrated by the dashed line named 'Maximum Outflow' in Figure 4.2. Note that $k_i^{\mathrm{c,free}} \geq 2$ to guarantee CFL conditions. In the case that link i is at an exit of the network, a constraint is introduced to limit the maximum outflow $q_i^{\mathrm{out,max}}(k^{\mathrm{c}})$ (veh/h) out of that link:

$$b_i^{\mathrm{L,eff}}(k^{\mathrm{c}}) q_i^{\mathrm{sat}} \leq q_i^{\mathrm{out,max}}(k^{\mathrm{c}}) \, \forall i \in I^{\mathrm{Exit}} \,, \tag{4.4}$$

where the set I^{Exit} is the set of exit links. This maximum outflow is modeled as an external disturbance to the process so that, for instance, the impact of a (temporal) bottleneck on the traffic flow at an exit of the network can be included.

The cumulative inflow to link i is updated using:

$$\begin{aligned} N_i^{\mathrm{L,in}}(k^{\mathrm{c}}+1) = N_i^{\mathrm{L,in}}(k^{\mathrm{c}}) + \sum_{i^{\mathrm{us}} \in I_i^{\mathrm{in}}} \left(\eta_{i^{\mathrm{us}},i}(k^{\mathrm{c}}) b_i^{\mathrm{L,eff}}(k^{\mathrm{c}}) q_i^{\mathrm{sat}} T^{\mathrm{c}} \right) + \ldots \\ \sum_{j \in J_i^{\mathrm{in}}} \left(\eta_{j,i}(k^{\mathrm{c}}) b_j^{\mathrm{O,eff}}(k^{\mathrm{c}}) q_j^{\mathrm{cap}} T^{\mathrm{c}} \right), \end{aligned} \tag{4.5}$$

where the set I_i^{in} is the set of links directly upstream of link i and the set J_i^{in} is the set of origins directly upstream of link i. The fraction $\eta_{i^{\mathrm{us}},i}(k^{\mathrm{c}})$ indicates the turn fraction form link i^{us} to link i, and the fraction $\eta_{j,i}(k^{\mathrm{c}})$ (-) indicates the turn fraction form origin j to link i. In order to model upstream propagating waves, the following constraint is used:

$$\begin{aligned} N_i^{\mathrm{L,in}}(k^{\mathrm{c}}+1) \leq \gamma_i^{\mathrm{c,shock}} N_i^{\mathrm{L,out}}(k^{\mathrm{c}} - k_i^{\mathrm{shock}} + 2) + \ldots \\ (1 - \gamma_i^{\mathrm{c,shock}}) N_i^{\mathrm{L,out}}(k^{\mathrm{c}} - k_i^{\mathrm{shock}} + 1) + N_i^{\mathrm{max}} \end{aligned} \tag{4.6}$$

with the number of vehicles N_i^{max} (veh) the maximum number of vehicles that can fit in a link – i.e., the link length multiplied with the jam density – the number of time steps $k_i^{\mathrm{c,shock}} = \lceil t_i^{\mathrm{shock}} / T^{\mathrm{c}} \rceil$ (-) , and the fraction $\gamma_i^{\mathrm{c,shock}} = k_i^{\mathrm{c,shock}} - t_i^{\mathrm{shock}} / T^{\mathrm{c}}$ (-) the residual of a sampling time-step that the upstream propagating wave travel time t_i^{shock} (h) is exceeded by $k_i^{\mathrm{c,shock}}$. It should hold that $k_i^{\mathrm{c,shock}} \geq 2$ in order to guarantee CFL conditions. This constraint limits the inflow as indicated with the dashed line named 'Maximum Inflow' in Figure 4.2.

Origins are modelled as vertical queues as illustrated in Figure 4.1. The cumulative inflow $N_j^{\mathrm{O,in}}(k^{\mathrm{c}})$ to origin j is updated as follows:

$$N_j^{\mathrm{O,in}}(k^{\mathrm{c}}+1) = N_j^{\mathrm{O,in}}(k^{\mathrm{c}}) + q_j^{\mathrm{in}}(k^{\mathrm{c}}) T^{\mathrm{c}} \,. \tag{4.7}$$

The cumulative outflow $N_j^{\mathrm{O,out}}(k^{\mathrm{c}}+1)$ out of origin j is updated using:

$$N_j^{\mathrm{O,out}}(k^{\mathrm{c}}+1) = N_j^{\mathrm{O,out}}(k^{\mathrm{c}}) + b_j^{\mathrm{O,eff}}(k^{\mathrm{c}}) q_j^{\mathrm{cap}} T^{\mathrm{c}} \,, \tag{4.8}$$

which should satisfy:

$$N_j^{\mathrm{O,out}}(k^{\mathrm{c}}+1) \leq N_j^{\mathrm{O,in}}(k^{\mathrm{c}}+1)\,. \tag{4.9}$$

Apart from these dynamical update equations and constraints, constraints on the control signals should be added:

$$0 \leq b_i^{\mathrm{L,eff}}(k^{\mathrm{c}}) \leq 1\,, \tag{4.10}$$

$$0 \leq b_j^{\mathrm{O,eff}}(k^{\mathrm{c}}) \leq 1\,, \tag{4.11}$$

$$\sum_{i \in I_y^{\mathrm{conflict}}} b_i^{\mathrm{L,eff}}(k^{\mathrm{c}}) \leq 1\,, \tag{4.12}$$

where the set I_y^{conflict} is the set y of signals which are in conflict with each other. The first two constraints make sure that the effective fractions are bounded between 0 and 1 while the third constraint makes sure that the green time is distributed over conflicting links. Note that clearance times between conflicts may be modeled by limiting the sum of the green-fractions of the conflicting links to be less than 1. These constraints essentially serve as the node model. To see this, note that in the original LTM model of Yperman [2007] the green times that are given may result in a violation of constraints (4.3) and (4.6). Hence, a node model is required to determine the transition flows from one link to another so that the constraints are not violated. In the approach proposed in this paper, the model is re-written as an optimization model where the effective fractions of green time are optimized so that the constraints (4.3) and (4.6) cannot be violated. Note that this can be seen as a modification of the generic class of first order node models proposed by Tampère et al. [2011] where instead of maximizing the node outflows, the total network outflows are maximized subject to supply and demand constraints of the nodes.

The state $x_i^{\mathrm{c,L}}(k^{\mathrm{c}}) \in \mathbb{R}^{n_i^{\mathrm{L,s}},1}$ of link i is given as:

$$x_i^{\mathrm{c,L}}(k^{\mathrm{c}}) = \ldots \tag{4.13}$$
$$\begin{bmatrix} N_i^{\mathrm{L,out}}(k^{\mathrm{c}}) & \ldots & N_i^{\mathrm{L,out}}(k^{\mathrm{c}}-k_i^{\mathrm{c,shock}}) & N_i^{\mathrm{L,in}}(k^{\mathrm{c}}) & \ldots & N_i^{\mathrm{L,in}}(k^{\mathrm{c}}-k_i^{\mathrm{c,free}}) \end{bmatrix}^{\top}$$

where $n_i^{\mathrm{L,s}} = k_i^{\mathrm{c,shock}} + k_i^{\mathrm{c,free}} + 2$ is the length of the vector. Similarly, the state $x_j^{\mathrm{c,O}}(k^{\mathrm{c}}) \in \mathbb{R}^{n_j^{\mathrm{O,s}},1}$ of an origin has the following structure:

$$x_j^{\mathrm{c,O}}(k^{\mathrm{c}}) = \begin{bmatrix} N_j^{\mathrm{O,out}}(k^{\mathrm{c}}) & N_j^{\mathrm{O,in}}(k^{\mathrm{c}}) \end{bmatrix}^{\top}, \tag{4.14}$$

where $n_j^{\mathrm{O,s}} = 2$ is the length of the vector.

4.2.3 Linear optimization problem formulation

The model described in the previous section consists of linear state equations and constraints. This sections details how these linear state equations and constraints can be

included in an optimization problem so that the network throughput can be optimized while considering all the traffic regimes.

The objective of the optimization is maximizing the network throughput. Under the assumption that the network inflow is not affected by the control action, maximizing throughput can be realized by minimizing the total time spent (TTS) used by all the vehicles in the network over the prediction horizon $K^{\mathrm{p}}T^{\mathrm{c}}$ (h). This is equivalent to minimizing the difference between the cumulative inflow and outflow of every link and origin in the network as represented using:

$$J(k^{\mathrm{m}},x)=\sum_{k^{\mathrm{c}}=(k^{\mathrm{m}}-1)\epsilon^{\mathrm{c,m}}+2}^{(k^{\mathrm{m}}-1)\epsilon^{\mathrm{c,m}}+K^{\mathrm{p}}} T^{\mathrm{c}}\Bigg\{\sum_{i\in I^{\mathrm{L}}}\left(N_i^{\mathrm{L,in}}(k^{\mathrm{c}})-N_i^{\mathrm{L,out}}(k^{\mathrm{c}})\right)+\ldots \\ \sum_{j\in I^{\mathrm{O}}}\left(N_j^{\mathrm{O,in}}(k^{\mathrm{c}})-N_j^{\mathrm{O,out}}(k^{\mathrm{c}})\right)\Bigg\}, \tag{4.15}$$

with I^{L} the set of all links, and I^{O} the set of all origins in the network.

This objective function can be formulated as a linear optimization problem of the following form:

$$\min_{\bar{u}(k^{\mathrm{m}})} Z\tilde{B}(k^{\mathrm{m}})\bar{u}(k^{\mathrm{m}})+Z\left(\tilde{A}x(k^{\mathrm{m}})+\tilde{C}\bar{d}(k^{\mathrm{m}})\right), \tag{4.16}$$

$$\text{Subject to} \tag{4.17}$$

$$M^{\mathrm{ineq}}(k^{\mathrm{m}})\bar{u}(k^{\mathrm{m}})\leq V^{\mathrm{ineq}}, \tag{4.18}$$

where the vector $\bar{u}(k^{\mathrm{m}})$ contains all the inputs that should be optimized, the vector $x(k^{\mathrm{m}})$ the state at time-step k^{m}, and the vector $\bar{d}(k^{\mathrm{m}})$ a prediction of the demand. The vector Z adds all the differences between cumulative inflows and cumulative outflows, and the matrices $\tilde{A}$, $\tilde{B}(k^{\mathrm{m}})$, and $\tilde{C}$ are used to compute the prediction of the states $\bar{x}(k^{\mathrm{m}})$ as specified in Section 4.2.3. The matrix $M^{\mathrm{ineq}}(k^{\mathrm{m}})$ and the vector V^{ineq} contain the inequality constraints as specified in Section 4.2.3.

By applying the receding horizon principle to this optimization problem an MPC strategy is obtained. The main concept of MPC is to find the optimal control signals $b_i^{\mathrm{L,eff}}(k^{\mathrm{c}})$ for time steps $k^{\mathrm{c}}=(k^{\mathrm{m}}-1)\epsilon^{\mathrm{c,m}}+1,\ldots,(k^{\mathrm{m}}-1)\epsilon^{\mathrm{c,m}}+K^{\mathrm{p}}-1$ by minimizing (4.15) over the prediction-horizon from $(k^{\mathrm{m}}-1)\epsilon^{\mathrm{c,m}}+2,\ldots(k^{\mathrm{m}}-1)\epsilon^{\mathrm{c,m}}+K^{\mathrm{p}})$ given the traffic state at time $t=k^{\mathrm{m}}T^{\mathrm{m}}$ and a prediction of the future disturbances $q^{\mathrm{in}}(k^{\mathrm{m}})$ and $q^{\mathrm{out,max}}(k^{\mathrm{m}})$. For the time steps $k=(k^{\mathrm{m}}-1)\epsilon^{\mathrm{c,m}}+1,\ldots,(k^{\mathrm{m}})\epsilon^{\mathrm{c,m}}$ the control signal applied to the process is defined as:

$$b_i(k)=b_i^{\mathrm{L,eff}}(\lfloor(k-1)/\epsilon^{\mathrm{c}}\rfloor+1), \tag{4.19}$$

where the mathematical operator $\lfloor\cdot\rfloor$ rounds the argument of the function to the nearest integer that is lower than the argument of the function. Next, the procedure is repeated at the next time step $k^{\mathrm{m}}T^{\mathrm{m}}+T^{\mathrm{m}}$ when new measurements become available.

Specification of linear objective function

The linear objective function consists of the matrices $\tilde{A}$, $\tilde{B}(k^{\mathrm{m}})$, and $\tilde{C}$ and the vector Z. To formulate these, it is required to write the model described in Section 4.2.2 in the standard linear form:

$$x(k^{\mathrm{c}}+1) = Ax(k^{\mathrm{c}}) + B(k^{\mathrm{c}})u(k^{\mathrm{c}}) + Cd(k^{\mathrm{c}})\,, \tag{4.20}$$

with the state $x(k^{\mathrm{c}}) \in \mathbb{R}^{n^{\mathrm{states}},1}$ given as:

$$x^{\mathrm{L}}(k^{\mathrm{c}}) = \begin{bmatrix} x_1^{\mathrm{L}}(k^{\mathrm{c}}) & \dots & x_{n^{\mathrm{L}}}^{\mathrm{L}}(k^{\mathrm{c}}) & x_1^{\mathrm{O}}(k^{\mathrm{c}}) & \dots & x_{n^{\mathrm{O}}}^{\mathrm{O}}(k^{\mathrm{c}}) \end{bmatrix}^{\top}, \tag{4.21}$$

where the number n^{L} (-) is the number of links, and the number n^{O} (-) is the number of origins. The number $n^{\mathrm{states}} = \sum_{i\in I^{\mathrm{L}}} n_i^{\mathrm{L,s}} + \sum_{j\in I^{\mathrm{O}}} n_j^{\mathrm{O,s}}$ denotes the length of this vector. The matrix $A \in \mathbb{R}^{n^{\mathrm{states}},n^{\mathrm{states}}}$ is defined in 4.A. The input vector $u(k^{\mathrm{c}}) \in \mathbb{R}^{n^{\mathrm{inputs}},1}$ is given by:

$$u(k^{\mathrm{c}}) = \begin{bmatrix} b_1^{\mathrm{L,eff}}(k^{\mathrm{c}}) & \dots & b_{n^{\mathrm{L}}}^{\mathrm{L,eff}}(k^{\mathrm{c}}) & b_1^{\mathrm{O,eff}}(k^{\mathrm{c}}) & \dots & b_{n^{\mathrm{O}}}^{\mathrm{O,eff}} \end{bmatrix}^{\top}, \tag{4.22}$$

with $n^{\mathrm{inputs}} = n^{\mathrm{L}} + n^{\mathrm{O}}$ the number of inputs. Finally, the disturbance vector $d(k^{\mathrm{c}}) \in \mathbb{R}^{n^{\mathrm{O}},1}$ is given by:

$$d(k^{\mathrm{c}}) = \begin{bmatrix} q_1^{\mathrm{in}}(k^{\mathrm{c}}) & \dots q_{n^{\mathrm{O}}}^{\mathrm{in}}(k^{\mathrm{c}}) \end{bmatrix}^{\top}. \tag{4.23}$$

Note that for an arbitrary time step $k^{\mathrm{c}} = k_o^{\mathrm{c}} + n$ with $k_o^{\mathrm{c}} = (k^{\mathrm{m}}-1)\epsilon^{\mathrm{c,m}} + 1$ (-) the state $x(k_0^{\mathrm{c}}+n)$ is given as:

$$x(k^{\mathrm{c}}+n) = A^n x(k^{\mathrm{c}}) + \sum_{i=1}^{n} A^{n-i}\Big(B(k^{\mathrm{c}}+i-1)u(k^{\mathrm{c}}+i-1) + Cd(k^{\mathrm{c}}+i-1)\Big)\,. \tag{4.24}$$

By stacking the predicted states at every time step $x(k_0^{\mathrm{c}}+n)$ from time step $k_0^{\mathrm{c}}+1$ to $k_o^{\mathrm{c}}+K^{\mathrm{p}}$ in vector $\bar{x}(k^{\mathrm{m}}) \in \mathbb{R}^{K^{\mathrm{p}}n^{\mathrm{states}},1}$:

$$\bar{x}(k^{\mathrm{m}}) = \begin{bmatrix} x_{k_0^{\mathrm{c}}+1} & \dots & x_{k_0^{\mathrm{c}}+K^{\mathrm{p}}} \end{bmatrix}^{\top}, \tag{4.25}$$

a prediction of the evolution of the states can be computed using following linear equation:

$$\bar{x}(k^{\mathrm{m}}) = \tilde{A}x(k^{\mathrm{m}}) + \tilde{B}(k^{\mathrm{m}})\bar{u}(k^{\mathrm{m}}) + \tilde{C}\bar{d}(k^{\mathrm{m}})\,. \tag{4.26}$$

The vector $\bar{u}(k^{\mathrm{m}}) \in \mathbb{R}^{n^{\mathrm{in,tot}},1}$ – with $n^{\mathrm{in,tot}} = K^{\mathrm{p}}(n^{\mathrm{L}}+n^{\mathrm{O}})$ – is defined as:

$$\bar{u}(k^{\mathrm{m}}) = \begin{bmatrix} u(k_0^{\mathrm{c}}) & \dots u(k_0^{\mathrm{c}}+K^{\mathrm{p}}-1) \end{bmatrix}^{\top}, \tag{4.27}$$

and the vector $\bar{d}(k^{\mathrm{m}}) \in \mathbb{R}^{n^{\mathrm{O}}K^{\mathrm{p}},1}$ is defined as:

$$\bar{d}(k^{\mathrm{m}}) = \begin{bmatrix} d(k_0^{\mathrm{c}}) & \dots d(k_0^{\mathrm{c}}+K^{\mathrm{p}}-1) \end{bmatrix}^{\top}, \tag{4.28}$$

The matrix $\tilde{A} \in \mathbb{R}^{n^{\text{states}}, n^{\text{states}}}$ is defined as:

$$\tilde{A} = \begin{bmatrix} A & A^2 & \dots & A^{K^{\text{p}}} \end{bmatrix}^\top , \tag{4.29}$$

the matrix $\tilde{B}(k^{\text{m}}) \in \mathbb{R}^{n^{\text{states}} K^{\text{p}}, n^{\text{in,tot}}}$ is defined as:

$$\tilde{B}(k^{\text{m}}) = \begin{bmatrix} B(k^{\text{c}}) & 0 & \dots & 0 \\ AB(k^{\text{c}}) & B(k^{\text{c}}+1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{K^{\text{p}}-1}B(k^{\text{c}}) & A^{K^{\text{p}}-2}B(k^{\text{c}}+1) & \dots & B(k^{\text{c}}+K^{\text{p}}-1) \end{bmatrix} , \tag{4.30}$$

(4.31)

and the matrix $\tilde{C} \in \mathbb{R}^{n^{\text{states}} K^{\text{p}}, n^{\text{O}} K^{\text{p}}}$ is defined as:

$$\tilde{C} = \begin{bmatrix} C & 0 & \dots & 0 \\ AC & C & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{K^{\text{p}}-1}C & A^{K^{\text{p}}-2}C & \dots & C \end{bmatrix} , \tag{4.32}$$

The vector $Z \in \mathbb{R}^{1, K^{\text{p}} n^{\text{states}}}$ is used to compute the value of the objective function by multiplication with $\tilde{A}x(k^{\text{m}}) + \tilde{B}(k^{\text{m}})\bar{u}(k^{\text{m}}) + \tilde{C}\bar{d}(k^{\text{m}})$. The vector Z is defined in 4.A.

Specification of linear constraints

Several constraints are included in the matrix $M^{\text{ineq}}(k^{\text{m}})$ and vectors V^{ineq} in (4.16). The matrix $M^{\text{ineq}}(k^{\text{m}})$, given as:

$$M^{\text{ineq}}(k^{\text{m}}) = \begin{bmatrix} M_1^{\text{ineq}}(k^{\text{m}}) & M_2^{\text{ineq}}(k^{\text{m}}) & M_3^{\text{ineq}}(k^{\text{m}}) & M_4^{\text{ineq}} & M_5^{\text{ineq}} & M_6^{\text{ineq}} & M_7^{\text{ineq}} \end{bmatrix}^\top , \tag{4.33}$$

and vector V^{ineq}

$$V^{\text{ineq}} = \begin{bmatrix} V_1^{\text{ineq}} & V_2^{\text{ineq}} & V_3^{\text{ineq}} & V_4^{\text{ineq}} & V_5^{\text{ineq}} & V_6^{\text{ineq}} & V_7^{\text{ineq}} \end{bmatrix}^\top . \tag{4.34}$$

consist of several parts, which make sure that the traffic flow modeling is in accordance with the LTM:

- The first part is used to model the free-flow dynamics according to (4.3).
- The second part is used to the spillback dynamics according to (4.6).
- The third part is used to constrain the outflow out of an origin according to (4.9).
- The fourth part is used to include the constraints (4.4) on the maximum outflow of the number n^{E} of exits in the network
- The fifth and sixth part are used to limit the control signals according to (4.10) and (4.11).
- The seventh part takes care of the conflicts (4.12).

Appendix 4.B provides a full description of these parts.

4.2.4 Dimension of the optimization problem

The dimension of the optimization problem influences the computation time required to solve it. This dimension is determined by the size of the input vector and of the constraint vector. The size of the input vector $\bar{u}(k^{\mathrm{m}})$ is $(n^{\mathrm{L}} + n^{\mathrm{O}})K^{\mathrm{p}}$. Additionally, a total number of $(4n^{\mathrm{L}} + 3n^{\mathrm{O}} + n^{\mathrm{E}} + n^{\mathrm{con}})K^{\mathrm{p}}$ inequality constraints are required, where n^{con} is the number of conflicts between links.

4.3 Simulation

The controller is evaluated using simulation in order to assess its behavior and performance. The indicators that are used to assess the performance of the control strategy are the TTS and the computation time used by the controller. The simulations are carried out in four steps:

1. Studying the qualitative behavior of the controller. The objective is to analyze whether the controller adequately responds to the different traffic regimes. To this end, a simple network and demand pattern are used so that it can be studied whether the computed control action is in accordance with expectations (addressed in Section 4.3.2).

2. Studying the quantitative performance of the controller. The objective is to study the performance of the controller in terms of realized TTS and computation time used. To this end, the controller is compared to two other, comparable strategies and the performance of these controllers when applied to a simple network and different demand patterns (see Section 4.3.3).

3. Studying the impact of the controller sampling time on the performance. To this end, simulations are carried out for different prediction horizons and controller sampling time steps (see Section 4.3.4).

4. Analyzing the application of the controller to a larger network. The objective is to study and compare the computation time required by the controller when the network size is increased. To this end, the three controllers are applied to a large network (for more details see Section 4.3.5).

4.3.1 Simulation set-up

The overall simulation set-up is detailed in Figure 4.3. The cell-transmission model (CTM) of Daganzo [1995] is chosen as the simulation model in combination with a demand-proportional node model. The simulation model is used as the ‘real-world’ situation to which the control signal is applied. The controller has an exact prediction

of the disturbances – i.e., the demand, outflow limitations, and turn fractions – available. The optimized green-fractions are directly applied to the CTM. The simulation sampling time step of the CTM is set to 1 second while the sampling time step of the prediction models are set to 10 seconds. The prediction horizon is set to 60 time steps (600 s), and the control signal is recomputed every 60 seconds.

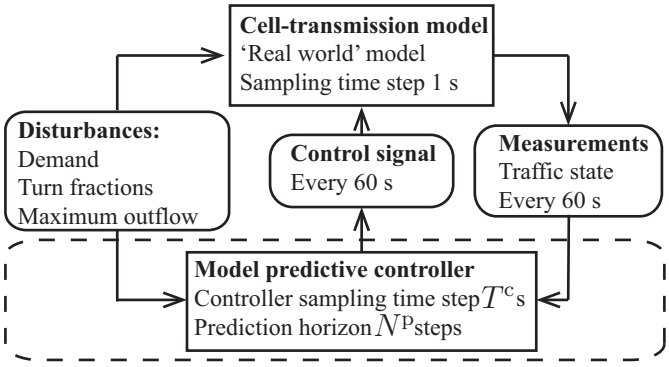


Figure 4.3: Overview of the simulation set-up. The default value of the prediction model sampling time T^c is 10 seconds and the default value of the prediction horizon N^p is 60 steps.

The characteristics of the link dynamics are determined by the free-flow speed which is set to 10 m/s, the upstream propagating wave speed which is set to -5 m/s, the jam density which is set to 400 veh/km, the saturation rate which is set to 2000 veh/h, and the segment length used in the CTM is set to 10 meters. In the different evaluations the length of links and the network structure are altered.

The simulations are carried out using Matlab R2015a on a computer with a 3.6 GHz processor and 16 Gb RAM. The linear optimization is carried out using the 'dual-simplex' algorithm implemented in the standard linear optimization function 'linprog' of Matlab. The computation time reported here consists of the computation time utilized by the optimization function at every controller time step.

4.3.2 Implementation of the control strategy: analyzing the qualitative behavior

The purpose of the first evaluation is to analyze the qualitative behavior. More specifically, the purpose is to study whether the controller is capable of reducing the outflow of the correct link when spillback is occurring. To this end, a simple network – called network 1 – as illustrated in Figure 4.4 is used for the evaluation. In this situation, the maximum outflow of link 3 is reduced to 600 veh/h. The length of each link is set to 200 meters, except for link 6 which has a length of 400 meters.

The simulation horizon was set to 3600 seconds. The demand pattern – i.e., demand pattern 1 in Figure 4.5 (A) – and turn fractions – of Table 4.1 – used for this evaluation are chosen to represent all traffic regimes. The network and demand pattern are chosen

Table 4.1: The turn fractions used in network I.

Turn fractions				
$\eta_{1,2} = 0.78$	$\eta_{2,3} = 0.4$	$\eta_{4,1} = 0.73$	$\eta_{6,3} = 0.6$	$\eta_{8,6} = 0.56$
$\eta_{1,5} = 0.22$	$\eta_{2,7} = 0.6$	$\eta_{4,5} = 0.27$	$\eta_{6,7} = 0.4$	$\eta_{8,9} = 0.44$

in such a way that the behavior of the controller can be interpreted. The first 450 seconds of the demand pattern represents the undersaturated regime. After time 450 s until time 1800 s the demand increases so that the flow towards the bottleneck exceeds its capacity. After time 1800 s the demand decreases again.

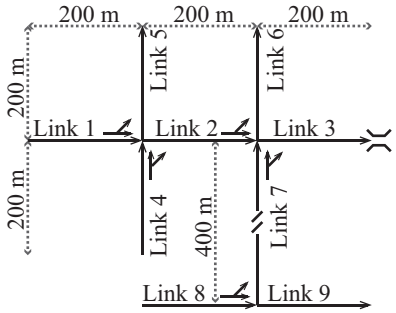


Figure 4.4: Network 1, a simple network.

Figure 4.6 and Figure 4.7 show the qualitative behavior of the controller. Figure 4.6 a shows the outflows of links 2, 3, 6, and 9 over time, Figure 4.6 b shows the number of vehicles in links 2, 3, and 6 – note that this is not the same as the queue length – and Figure 4.7 shows snapshots of the number of vehicles in every link and the link outflows at different time instances. Using these figures, the qualitative behavior of the controller is studied below.

- During the first 450 seconds the traffic situation is undersaturated. From Figure 4.6 A it can be observed that it takes some time before the flow reaches the links.

- After time 450 s the demand increases and the capacity of the bottleneck at link 3 is exceeded. This causes the number of vehicles in link 3 to increase. The number of vehicles in link 6 also starts to increase, since, the combined demand of link 2 and link 6 is approximately 2500 veh/h and the turn fraction from 2 to 7 is larger compared to the turn fraction from link 6 to 7, so the controller gives priority to link 2. In Figure 4.7 B the number of vehicles in links 3 and 6 have increased considerably and the arrow indicates that these are increasing.

- Around time 1350 s link 3 is full and the controller reduces the outflow of link 6 to 0 veh/h. The flow from link 2 to link 3 is then exactly 600 veh/h so that the inflow to link 3 is equal to its outflow. The outflow of link 7 is then 900 veh/h. This situation is illustrated in Figure 4.7 C.

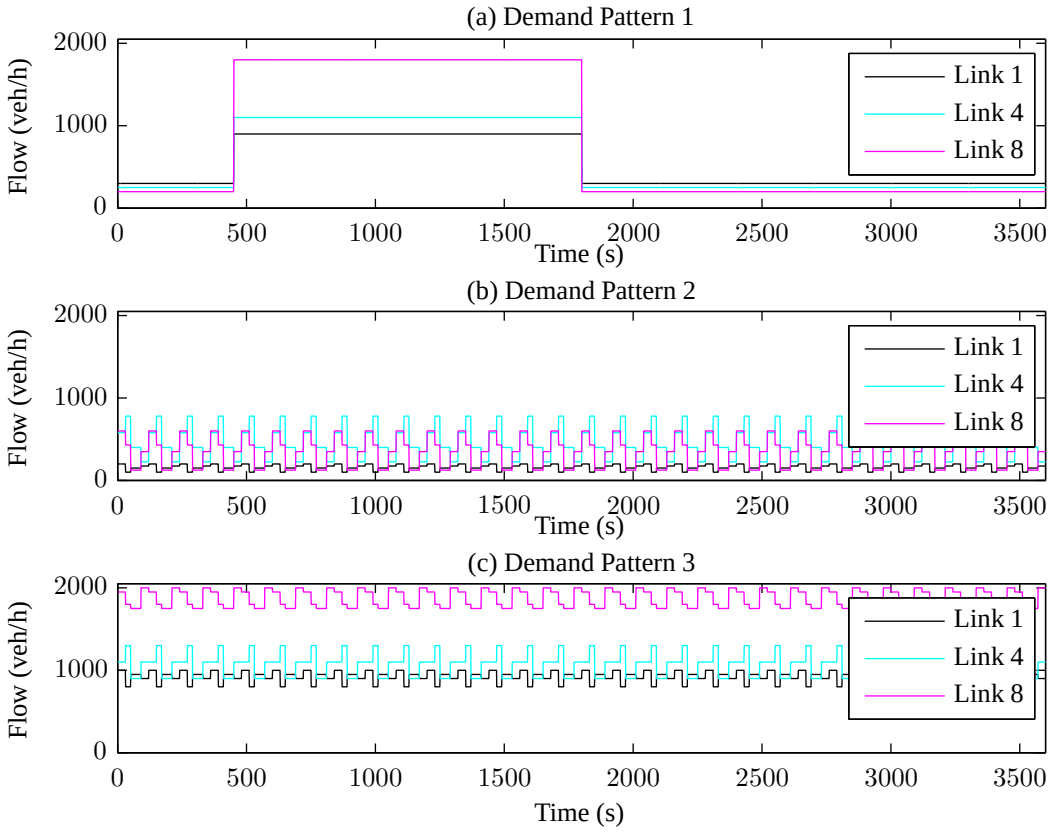


Figure 4.5: Demand patterns applied to network I, (A) pattern I, all traffic regimes, (B) pattern II, the undersaturated regime, (C) pattern III, saturated and oversaturated regimes.

- At time 1390 s link 6 is full as well and now the outflow of link 6 is increased to 1000 veh/h so that the queue does not spillback to link 8 and blocking of link 9 is prevented. Simultaneously, the outflow from link 2 is reduced to 0 veh/h so that the number of vehicles in this queue starts to increase. This causes the outflow of link 9 to be preserved at 800 veh/h while the flow out of link 7 is reduced to 400 veh/h. These effects can be observed in Figure 4.7 D at time 1500 s.
- Link 2 is full around time 1520 seconds. At that time, the controller reduces the outflow from link 6 to 0 veh/h which causes spillback to link 8 but prevents spillback to links 1 and 5. This spillback reduces the outflow from link 9 from 800 veh/h to 0 veh/h and increases the outflow of link 7 to 900 veh/h and preserves the outflow of link 5 at 500 veh/h as illustrated in Figure 4.7 E.
- At time 1880 seconds the flow out of link 9 increases again, since then the demand has decreased again and the flow out of link 6 is increased as well. A snapshot of the network at time 2300 s is shown in Figure 4.7 F.

Another observation that can be made from Figure 4.6 B is that the maximum number of vehicles that fits in a link changes over time. For instance, the maximum number of vehicles in link 2 at time 1200 s is smaller than the maximum number of vehicles in link 2 at time 1530 s. The reason for this is that the smaller the link outflow, the less voids between vehicles have to propagate through the link so more vehicles can be present in the link. This behavior is not included in most linear MPC approaches that use other models.

Summarizing, this evaluation shows that the controller acts as expected. It is capable of considering free-flow dynamics, and take the impact of spillback into account. Most importantly, it is capable of modelling the effect that the maximum storage space in a link is influenced by the link outflow due to upstream propagating waves.

4.3.3 Comparative evaluation: quantitative analysis of the controller performance

The second evaluation is carried out to analyze the quantitative performance. To this end, the approach is compared to two other, comparable MPC approaches, namely, the approach of Aboudolas et al. [2010] and of Le et al. [2013]. The reason why these approaches are chosen are that they are both of the linear MPC type, aggregate the traffic dynamics to several (tens) of seconds, and assume that the turn fractions are known. The main differences are that they exploit other prediction models. The approach of Aboudolas et al. [2010] is especially designed for (over) saturated regimes, hence, it does not consider free-flow travel times. The approach of Le et al. [2013] does consider free-flow travel times. Both the approaches of Aboudolas et al. [2010] and Le et al. [2013] do not include the upstream propagating waves caused by spill back. Thus, it is expected that in undersaturated regimes the method of Le et al. [2013] and

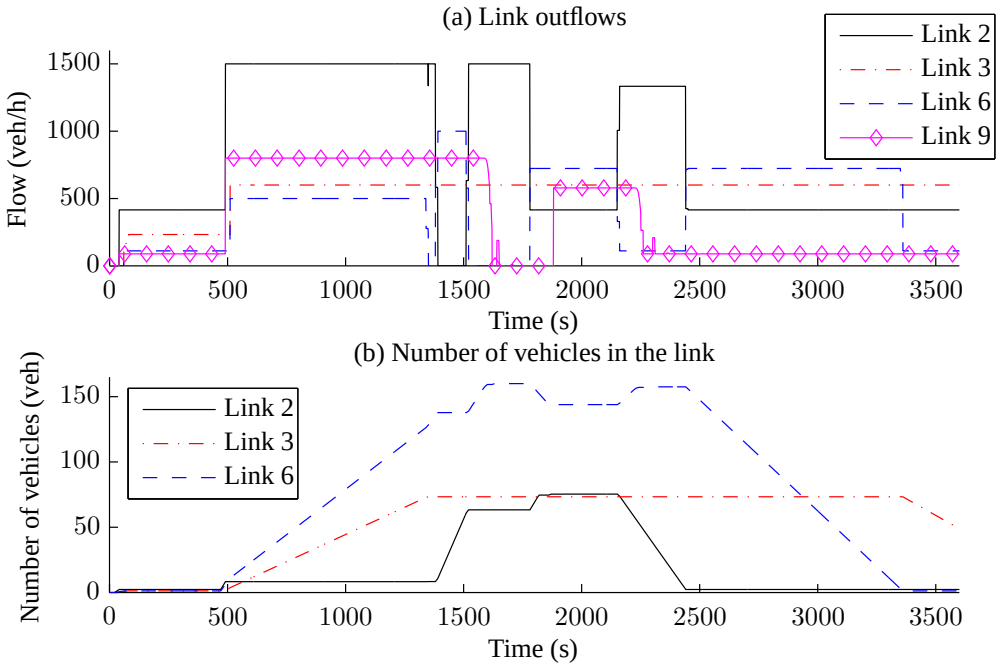


Figure 4.6: (a) The outflows out of links 2, 3, 6, and 9 over time. (b) The number of vehicles in links 2, 3, and 6 over time.

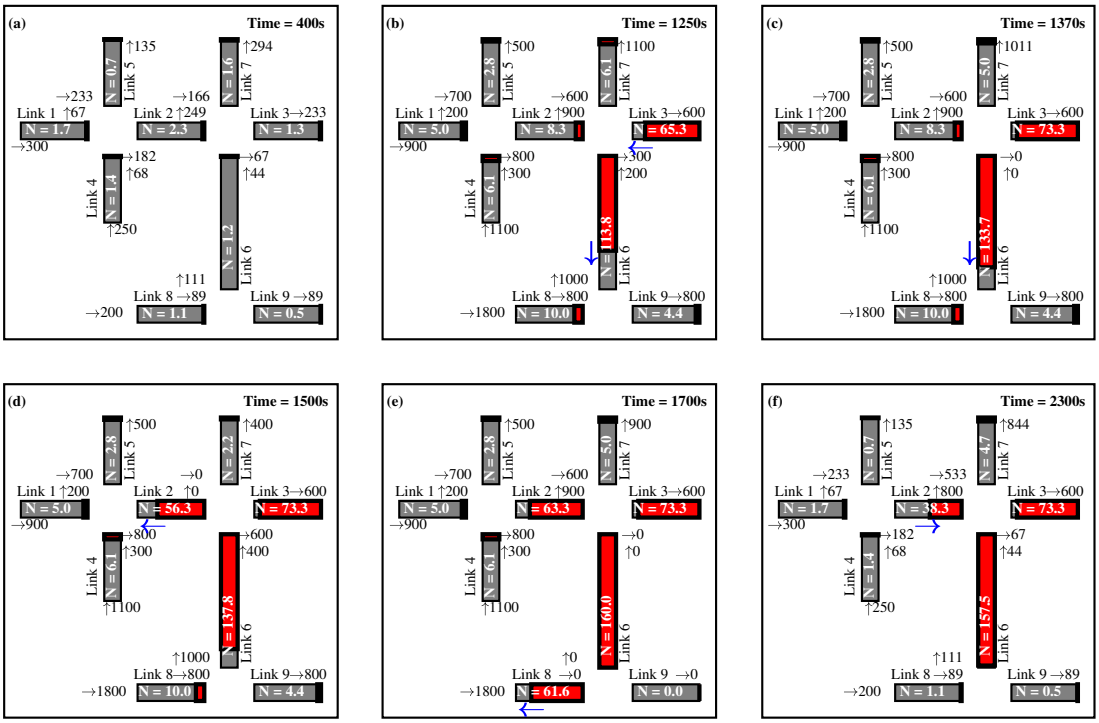


Figure 4.7: Snapshots of the network state at different time instances. The red bars indicate the number of vehicles in the link, not the queue length. (a) the undersaturated regime at time 400 s. (b) around time 1250 s the number of vehicles in links 3 and 6 grow. (c) around time 1370 s link 3 is full and the flow out of link 6 is reduced to 0 veh/h. (d) around time 1500 s link 6 is full and the flow out of link 2 is reduced to 0 veh/h. (e) around time 1700 s link 2 is full and the flow out of link 6 is reduced to 0 veh/h. (f) the demand decreases after time 1800 s.

the approach proposed in this paper achieve similar performance in terms of TTS. In oversaturated regimes it is expected that the approach proposed in this paper can realize a lower TTS compared to the approaches of Aboudolas et al. [2010] and Le et al. [2013] because of the inclusion of the upstream propagating waves.

It must be noted that the objective functions exploited in [Aboudolas et al., 2010] and [Le et al., 2013] are different from the one presented in this paper. Therefore, the approaches of Aboudolas et al. [2010] and Le et al. [2013] are adopted to the objective function used in this paper. In this way, the main difference between the approaches is the prediction models used to formulate the optimization problem.

In order to test these expectations, network 1 is used with three different demand patterns as detailed in Figure 4.5. The first demand pattern contains all traffic regimes. The second demand pattern only contains undersaturated traffic regimes. To realize this, the bottleneck at the exit of link 3 is removed. The third demand pattern consists of saturated and oversaturated regimes. To obtain a fair comparison the network is saturated first by applying the control strategy proposed in this paper for this first 120 seconds and these first 120 seconds are removed from the TTS computations.

The quantitative results of the evaluation are presented in Table 4.2. It can be observed that in undersaturated regimes – i.e., demand pattern 2 – the method proposed in this paper realizes the same TTS as the approach of Le et al. [2013]. In that situation, the approach of Aboudolas et al. [2010] has a worse performance, since, it does not consider free-flow travel times. It can also be observed that in the saturated regime – i.e., demand pattern 3 –, the approach proposed in this paper has improved performance. The reason for this is that the controller considers the upstream propagating waves when determining the maximum link inflow. The method proposed in this paper can realize a lower TTS for the first demand pattern as well.

From Table 4.2 it can also be observed that the average computation times used by the approach proposed in this paper are below 0.25 seconds. The approach of *et al.* Aboudolas et al. [2010] has the lowest computation time even though the dimension of the optimization problem – i.e., 720 variables and 3060 inequality constraints – is the same as the dimension of the optimization problem proposed in this paper. The maximum computation time of the approach proposed by Le et al. [2013] is the largest. The reason for this is that a link is divided into segments – or classes – and for every class a dummy variable is added which has to be optimized resulting in 1380 variables and 6900 inequality constraints.

4.3.4 Impact of controller timing on performance

The next set of evaluations is conducted to analyze the controller performance when changing the prediction horizon, and controller sampling time step. It is expected that increasing the prediction horizon and decreasing the controller sampling time step will lead to a lower TTS but a higher computation time.

Table 4.2: Overview of the results comparing the average CPU time (ACPU) in seconds used by the optimization and the TTS in veh·h used by all the vehicles in the network for the different demand patterns.

Demand Pattern	Method	LML-U	[Aboudolas et al., 2010]	[Le et al., 2013]
1	ACPU	0.20	0.14	0.66
All regimes	TTS	184.59	186.87(+1.2%)	186.48(+1.0%)
2	ACPU	0.19	0.16	0.76
Undersaturated	TTS	15.48	17.68(+14.2%)	15.48(+0.0%)
3	ACPU	0.25	0.18	0.70
(Over)saturated	TTS	725.01	735.80(+1.5%)	736.87(+1.6%)

To this end, the LML-U control strategy is applied to network 1 and demand pattern 1 with different combinations of prediction horizon and prediction model sampling time step. The results are presented in Table 4.3. The table shows that increasing the prediction horizon to 300 seconds results in a lower TTS. A further increase does not lead to a lower TTS. The reason for this is that the prediction horizon should be long enough to include all relevant dynamics, such as, forward and backward propagating waves. A horizon of 300 seconds is thus long enough to anticipate the impact of the control actions on the network outflow.

It can also be observed that increasing the prediction model sampling time step or decreasing the prediction horizon reduces the required computation time. It can be observed that a time step of 10 seconds leads to a lower TTS when compared to a time step of 20 seconds. A time step of 2 of 5 seconds does not lead to a lower TTS when compared to a TTS of 10 seconds. This is probably caused by the demand pattern which is rather constant so that there is no need to consider dynamics with a resolution that is higher than 10 seconds.

4.3.5 Application of the controller to a large network

The fourth evaluation is conducted to test the controller when applied to large networks. To this end, network 2 as illustrated in Figure 4.8 is used for the simulation network. This network consist of 80 links with varying link lengths. Bottlenecks with a capacity of 300 veh/h are placed at the exits of links 5, 30, 50, and 70. The turn fractions out of every link are set to 1/3 and the demand pattern is chosen identical for every link, namely, 250 veh/h for the first 250 seconds, 800 veh/h from time 450 s to time 1800 s and 250 veh/h after time 1800 s.

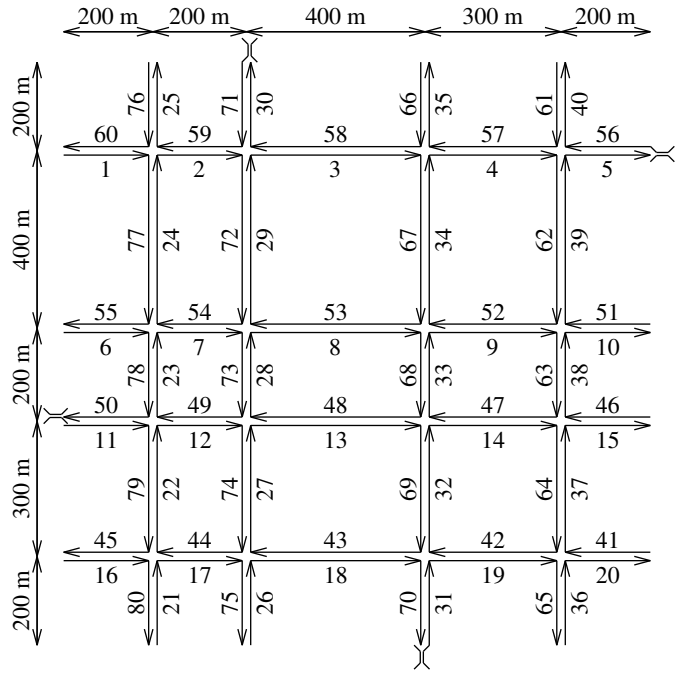


Figure 4.8: Network 2, a large grid network with varying link lengths

Table 4.3: Overview of the results comparing the average CPU time (ACPU) in seconds used by the optimization and the TTS in veh·h used by all the vehicles in the network for the different combinations of the prediction horizon and controller sampling time step T^c.

	T^c (s)							
	2 s		5 s		10 s		20 s	
Prediction horizon (s)	ACPU	TTS	ACPU	TTS	ACPU	TTS	ACPU	TTS
20	0.036	1161.07	0.024	1161.07	0.020	1161.07	0.022	1074.32
40	0.032	661.14	0.024	662.11	0.024	662.17	0.023	577.81
60	0.042	244.69	0.029	288.59	0.028	288.52	0.024	290.04
80	0.11	187.66	0.029	187.66	0.028	187.66	0.023	190.15
100	0.099	187.66	0.033	187.66	0.026	187.66	0.024	190.14
200	0.45	186.88	0.069	186.85	0.033	186.80	0.029	189.23
300	1.56	184.60	0.14	184.60	0.044	184.59	0.030	187.05
400	4.32	184.60	0.28	184.60	0.069	184.59	0.034	187.05
500	10.01	184.60	0.56	184.60	0.10	184.59	0.040	187.05
600	17.12	184.60	0.96	184.60	0.14	184.59	0.048	187.05
700	26.03	184.60	1.52	184.60	0.21	184.59	0.055	187.05
800	40.3	184.60	2.33	184.60	0.28	184.59	0.071	187.05
900	61.2	184.60	3.30	184.60	0.39	184.59	0.081	187.05

Table 4.4: Overview of the results comparing the average CPU time (ACPU), maximum CPU time (MCPU), and standard deviation (SD) of the CPU time in seconds used by the optimization and the TTS in veh·h used by all the vehicles in network 2.

Method	ACPU	MCPU	SD of CPU time	TTS
LML-U	45.4	58.9	6.1	1266.5
[Aboudolas et al., 2010]	17.8	25.5	3.1	1342.4(+6.0%)
[Le et al., 2013]	225.4	271.4	16.5	1278.6(+0.9%)

Two observations can be made from Table 4.4 which provides an overview of the results. First of all, compared to the approach of Aboudolas et al. [2010], a TTS gain of 6.1% is realized while requiring more CPU time. The reason for this is that both approaches model links as a single element. However, the dimension of the optimization problem proposed in this paper is higher, since, the initial traffic state is larger due to the fact that it includes the history to model downstream and upstream propagating waves. Thus, in saturated only regimes, the approach of Aboudolas et al. [2010] gives the best trade-off between computation time and controller performance. However, when applying a controller to all traffic regimes, the choice between both approaches depends on the network size, i.e., the LML-U approach can achieve a better throughput improvement but for real-time operation, the computation time should remain smaller than the controller sampling time. For instance, for this network of 80 links and 16 origins, the average CPU time of 45.9 seconds is still below the controller sampling time of 60 seconds thus the LML-U approach gives a better throughput improvement with reasonable CPU time. Secondly, compared to the approach of Le et al. [2013] a TTS gain of 1.0% is realized in considerably less CPU time. This shows that it is beneficial to consider a link as a single element instead of dividing a link into segments.

4.4 Conclusions and recommendations

A linear MPC strategy for the optimization of urban road network throughput in all traffic regimes was developed and evaluated in this paper. The main contribution of this paper is the formulation of a linear optimization problem which can be efficiently solved that considers queuing dynamics and downstream and upstream propagating waves. This was realized by describing the link dynamics using the link transmission model, and aggregating the traffic dynamics to (several) tens of seconds. Simulations were carried out to test the approach. A qualitative analysis of the controller performance indicated that the approach is capable of dealing with the impact of upstream propagating waves on queue spillback. More specifically, it has been shown that the controller can take the impact of the link outflow on the maximum link inflow into account. A quantitative comparison has been done by employing two comparable, linear MPC strategies. It has been found that the approach proposed in this paper can realize a better throughput in oversaturated regimes, due to the inclusion of upstream prop-

agating waves caused by spill back, when compared to the other approaches. The evaluations showed that in terms of controller performance and computation time, an optimization approach that considers a link as a single element, instead of dividing a link into segments, results in a better trade-off between computation time and controller performance. When compared to a store-and-forward-based approach, it was found that the proposed approach realizes a higher throughput but also requires a higher computation time.

Further research should be carried out to extend the approach to include detailed signal plans and to relax the assumption of known turn fractions. Also, more simulation-based studies should be carried out, utilizing more realistic traffic models. Additionally, the impact of measurement noise and uncertainties should be studied. Further mathematical analyses can be carried out to study certain controller properties, such as, scalability and stability. Finally, the impact of heterogeneous traffic on the controller performance and the inclusion of other objective functions in the optimization problem may be investigated in the future.

Acknowledgements

This work is part of the research programme 'The Application of Operations Research in Urban Transport', which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

4.A Specification of objective function matrices

This appendix details the matrices used in Section 4.2.3. First, the matrix A is specified. The matrix $A \in \mathbb{R}^{n^{\text{states}},n^{\text{states}}}$ is a matrix consisting of the matrices $A_i^{\text{L}} \in \mathbb{R}^{n_i^{\text{L,s}},n_i^{\text{L,s}}}$ and $A_j^{\text{O}} \in \mathbb{R}^{n_j^{\text{O,s}},n_j^{\text{O,s}}}$ of the links and origins respectively on its diagonal:

$$A^{\text{L}} = \begin{bmatrix} A_1^{\text{L}} & & & & & \\ & \ddots & & & & \\ & & A_{n^{\text{L}}}^{\text{L}} & & & \\ & & & A_1^{\text{O}} & & \\ & & & & \ddots & \\ & & & & & A_{n^{\text{O}}}^{\text{L}} \end{bmatrix}, \tag{4.35}$$

with the matrix A_i^{L} of a link i given by:

$$A_i^{\mathrm{L}} = \begin{bmatrix} 1 & & & & & & & \\ 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & 0 & & & & \\ & & & & 1 & & & \\ & & & & 1 & & & \\ & & & & & \ddots & & \\ & & & & & & 1 & 0 \end{bmatrix}, \tag{4.36}$$

and the matrix A_j^{O} of origin j given by:

$$A_j^{\mathrm{O}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{4.37}$$

Next, the matrix $B(k^{\mathrm{c}}) \in \mathbb{R}^{n^{\mathrm{states}}, n^{\mathrm{L}}+n^{\mathrm{O}}}$ is defined as:

$$B(k^{\mathrm{c}}) = \begin{bmatrix} B_1^{\mathrm{L}}(k^{\mathrm{c}}) & \dots & B_{n^{\mathrm{L}}}^{\mathrm{L}}(k^{\mathrm{c}}) & B_1^{\mathrm{O}}(k^{\mathrm{c}}) & \dots & B_{n^{\mathrm{O}}}^{\mathrm{O}}(k^{\mathrm{c}}) \end{bmatrix}^{\top}, \tag{4.38}$$

where the matrix $B_i^{\mathrm{L}}(k^{\mathrm{c}}) \in \mathbb{R}^{n_i^{\mathrm{L,s}}, n^{\mathrm{L}}+n^{\mathrm{O}}+n^{\mathrm{O}}}$ of link i is given as:

$$B_i^{\mathrm{L}}(k^{\mathrm{c}}) = \begin{bmatrix} B_{i,1}^{\mathrm{L}}(k^{\mathrm{c}}) & B_{i,2}^{\mathrm{L}}(k^{\mathrm{c}}) \end{bmatrix}, \tag{4.39}$$

with the matrix $B_{i,1}^{\mathrm{L}}(k^{\mathrm{c}}) \in \mathbb{R}^{n_i^{\mathrm{L,s}}, n^{\mathrm{L}}}$ given as:

$$B_{i,1}^{\mathrm{L}}(k^{\mathrm{c}}) = \begin{bmatrix} 0 & \dots & 0 & q_i^{\mathrm{sat}} T^{\mathrm{c}} & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \dots & \vdots & \vdots & \vdots & \ddots & \vdots \\ N_{1,i}^{\mathrm{T}} & \dots & N_{i-1,i}^{\mathrm{T}} & 0 & N_{i+1,i}^{\mathrm{T}} & \dots & N_{n^{\mathrm{L}},i}^{\mathrm{T}} \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{bmatrix}. \tag{4.40}$$

with $N_{j,i}^{\mathrm{T}} = \eta_{j,i}(k^{\mathrm{c}}) q_j^{\mathrm{sat}} T^{\mathrm{c}}$ and the matrix $B_{i,2}^{\mathrm{L}}(k^{\mathrm{c}}) \in \mathbb{R}^{n_i^{\mathrm{L,s}}, n^{\mathrm{O}}}$ defined as follows:

$$B_{i,2}^{\mathrm{L}}(k^{\mathrm{c}}) = \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ \eta_{1,i}(k^{\mathrm{c}}) q_{w,1}^{\mathrm{cap}} T^{\mathrm{c}} & \dots & \eta_{n^{\mathrm{O}},i}(k^{\mathrm{c}}) q_{n^{\mathrm{O}}}^{\mathrm{cap}} T^{\mathrm{c}} \\ 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix}. \tag{4.41}$$

The matrix $B_j^{\mathrm{O}}(k^{\mathrm{c}}) \in \mathbb{R}^{n_j^{\mathrm{O,s}}, n^{\mathrm{L}}+n^{\mathrm{O}}}$ of origin j is given as:

$$B_j^{\mathrm{O}}(k^{\mathrm{c}}) = \begin{bmatrix} B_{j,1}^{\mathrm{O}}(k^{\mathrm{c}}) & B_{j,2}^{\mathrm{O}}(k^{\mathrm{c}}) \end{bmatrix} , \tag{4.42}$$

with

$$B_{j,1}^{\mathrm{O}}(k^{\mathrm{c}}) = 0, \in \mathbb{R}^{n_j^{\mathrm{O,s}}, n^{\mathrm{L}}}, \tag{4.43}$$

and the matrix $B_{j,2}^{\mathrm{O}}(k^{\mathrm{c}}) \in \mathbb{R}^{n_j^{\mathrm{O,s}}, n^{\mathrm{O}}}$ defined as follows:

$$B_{j,2}^{\mathrm{O}}(k^{\mathrm{c}}) = \begin{bmatrix} 0 & \dots 0 & q_{w,j}^{\mathrm{cap}}(k^{\mathrm{c}}) T^{\mathrm{c}} & 0 & \dots & 0 \\ 0 & \dots 0 & 0 & 0 & \dots & 0 \end{bmatrix} . \tag{4.44}$$

The matrix $C \in \mathbb{R}^{n^{\mathrm{states}}, n^{\mathrm{O}}}$ is defined as follows:

$$C = \begin{bmatrix} C_1^{\mathrm{L}} & \dots & C_{n^{\mathrm{L}}}^{\mathrm{L}} & C_1^{\mathrm{O}} & \dots & C_{n^{\mathrm{O}}}^{\mathrm{O}} \end{bmatrix}^{\top} . \tag{4.45}$$

The matrix $C_i^{\mathrm{L}} = 0 \in \mathbb{R}^{n_i^{\mathrm{L,s}}, n^{\mathrm{O}}}$, since, there is no demand directly going into a link. The matrix $C_j^{\mathrm{O}} = 0 \in \mathbb{R}^{n_j^{\mathrm{O,s}}, n^{\mathrm{O}}}$ is given as:

$$C_j^{\mathrm{O}} = \begin{bmatrix} 0 & \dots 0 & 0 & 0 & \dots & 0 \\ 0 & \dots 0 & T^{\mathrm{c}} & 0 & \dots & 0 \end{bmatrix} . \tag{4.46}$$

Finally, the vector $Z \in \mathbb{R}^{1, n^{\mathrm{in,tot}}}$ is used to compute the value of the objective function as specified in (4.16). The vector Z is defined as follows:

$$Z = T^{\mathrm{c}} \begin{bmatrix} Z_k & \dots & Z_k \end{bmatrix} , \tag{4.47}$$

$$Z_k = T^{\mathrm{c}} \begin{bmatrix} Z_1^{\mathrm{L}} & \dots Z_{n^{\mathrm{L}}}^{\mathrm{L}} & Z_i^{\mathrm{O}} & \dots & Z_{n^{\mathrm{O}}}^{\mathrm{O}} \end{bmatrix} , \tag{4.48}$$

with the vector $Z_i^{\mathrm{L}} \in \mathbb{R}^{1, n_i^{\mathrm{L,s}}}$ of link i defined as:

$$Z_i^{\mathrm{L}} = \begin{bmatrix} -1 & 0 & \dots & 0 & 1 & - & \dots & 0 \end{bmatrix} , \tag{4.49}$$

and the vector $Z_j^{\mathrm{O}} \in \mathbb{R}^{1,2}$ of origin j defined as:

$$Z_j^{\mathrm{O}} = \begin{bmatrix} -1 & 1 \end{bmatrix} . \tag{4.50}$$

4.B Specification of inequality constraints

The first matrix $M_1^{\mathrm{ineq}}(k^{\mathrm{m}}) \in \mathbb{R}^{n^{\mathrm{L}} K^{\mathrm{p}}, n^{\mathrm{in,tot}}}$ and vector $V_1^{\mathrm{ineq}} \in \mathbb{R}^{n^{\mathrm{L}} K^{\mathrm{p}}, 1}$ are used to model the free-flow dynamics according to (4.3). This constraint is applied to the predicted state:

$$\bar{M}_1^{\mathrm{ineq}} \bar{x}(k^{\mathrm{m}}) \leq 0 , \tag{4.51}$$

$$\bar{M}_1^{\mathrm{ineq}} (\tilde{A} x(k^{\mathrm{m}}) + \tilde{B}(k^{\mathrm{m}}) \bar{u}(k^{\mathrm{m}}) + \tilde{C} \bar{d}(k^{\mathrm{m}})) \leq 0 , \tag{4.52}$$

$$\bar{M}_1^{\mathrm{ineq}} \tilde{B}(k^{\mathrm{m}}) \bar{u}(k^{\mathrm{m}}) \leq -\bar{M}_1^{\mathrm{ineq}} (\tilde{A} x(k^{\mathrm{m}}) + \tilde{C} \bar{d}(k^{\mathrm{m}})) . \tag{4.53}$$

So that:

$$M_1^{\text{ineq}}(k^{\text{m}}) = \bar{M}_1^{\text{ineq}} \tilde{B}(k^{\text{m}}) \,, \tag{4.54}$$

$$V_1^{\text{ineq}} = -\bar{M}_1^{\text{ineq}} (\tilde{A} x(k^{\text{m}}) + \tilde{C} \bar{d}(k^{\text{m}})) \,, \tag{4.55}$$

here, the matrix $\bar{M}_1^{\text{ineq}} \in \mathbb{R}^{n^{\text{L}} K^{\text{p}}, n^{\text{states}} K^{\text{p}}}$ is given as:

$$\bar{M}_1^{\text{ineq}} = \begin{bmatrix} \ddots & & 0 \\ & \begin{bmatrix} M^1 & 0 \end{bmatrix} & \\ 0 & & \ddots \end{bmatrix} , \tag{4.56}$$

with the matrix $M^1 \in \mathbb{R}^{n^{\text{L}}, n^{\text{states}}}$ given as:

$$M^1 = \tag{4.57}$$

$$\begin{bmatrix} \ddots & & & 0 \\ & \begin{bmatrix} 1 & 0 & \dots & 0 & -\gamma_i^{\text{c,free}} & -(1-\gamma_i^{\text{c,free}}) \end{bmatrix} & & \\ 0 & & & \ddots \end{bmatrix} .$$

The matrix $M_2^{\text{ineq}}(k^{\text{m}}) \in \mathbb{R}^{n^{\text{L}} K^{\text{p}}, n^{\text{in,tot}}}$ and matrix $V_2^{\text{ineq}} \in \mathbb{R}^{n^{\text{L}} K^{\text{p}}, 1}$ are used to model the spillback conditions according to (4.6). In a similar way as in (4.53) these are given as:

$$M_2^{\text{ineq}}(k^{\text{m}}) = \bar{M}_2^{\text{ineq}} \tilde{B}(k^{\text{m}}) \,, \tag{4.58}$$

$$V_2^{\text{ineq}} = \bar{V}_2^{\text{ineq}} - \bar{M}_2^{\text{ineq}} (\tilde{A} x(k^{\text{m}}) + \tilde{C} \bar{d}(k^{\text{m}})) \,, \tag{4.59}$$

here, the matrix $\bar{M}_2^{\text{ineq}} \in \mathbb{R}^{n^{\text{L}} K^{\text{p}}, n^{\text{states}} K^{\text{p}}}$ is given as:

$$\bar{M}_2^{\text{ineq}} = \begin{bmatrix} \ddots & & 0 \\ & \begin{bmatrix} M^2 & 0 \end{bmatrix} & \\ 0 & & \ddots \end{bmatrix} , \tag{4.60}$$

with the matrix $M^2 \in \mathbb{R}^{n^{\text{L}}, n^{\text{states}}}$ given as:

$$M^2 = \begin{bmatrix} \ddots & & & 0 \\ & \begin{bmatrix} 0 & \dots & 0 & -\gamma_i^{\text{c,shock}} & -(1-\gamma_i^{\text{c,shock}}) & 1 & 0 & \dots & 0) \end{bmatrix} & & \\ 0 & & & \ddots \end{bmatrix} . \tag{4.61}$$

(4.62)

The vector $\bar{V}_2^{\text{ineq}} \in \mathbb{R}^{n^{\text{L}} K^{\text{p}}, 1}$ is given as:

$$\bar{V}_2^{\text{ineq}} = \begin{bmatrix} \tilde{V}_2^{\text{ineq}} & \dots & \tilde{V}_2^{\text{ineq}} \end{bmatrix}^\top , \tag{4.63}$$

with $\tilde{V}_2^{\text{ineq}} \in \mathbb{R}^{n^{\text{L}},1}$

$$\tilde{V}_2^{\text{ineq}} = \begin{bmatrix} N_1^{\max} & \dots & N_{n^{\text{L}}}^{\max}, \end{bmatrix}^\top . \tag{4.64}$$

The third matrix $M_3^{\text{ineq}}(k^{\text{m}}) \in \mathbb{R}^{n^{\text{O}}K^{\text{p}},n^{\text{in,tot}}}$ and vector $V_3^{\text{ineq}} \in \mathbb{R}^{n^{\text{O}}K^{\text{p}},1}$ are used to constrain the outflow out of an origin according to (4.9) and are given as:

$$M_3^{\text{ineq}}(k^{\text{m}}) = \bar{M}_3^{\text{ineq}} \tilde{B}(k^{\text{m}}) , \tag{4.65}$$

$$V_3^{\text{ineq}} = -\bar{M}_3^{\text{ineq}} (\tilde{A}x(k^{\text{m}}) + \tilde{C}\bar{d}(k^{\text{m}})) , \tag{4.66}$$

here, the matrix $\bar{M}_3^{\text{ineq}} \in \mathbb{R}^{n^{\text{O}}K^{\text{p}},n^{\text{states}}K^{\text{p}}}$ is given as:

$$\bar{M}_3^{\text{ineq}} = \begin{bmatrix} \ddots & & & & 0 \\ & \begin{bmatrix} \ddots & & & 0 \\ 0 & & 1 & -1 & \\ & 0 & & & \ddots \end{bmatrix} & & & \\ 0 & & & & \ddots \end{bmatrix} . \tag{4.67}$$

The matrix $M_4^{\text{ineq}} \in \mathbb{R}^{n^{\text{E}}K^{\text{p}},n^{\text{in,tot}}}$ and vector $V_4^{\text{ineq}} \in \mathbb{R}^{n^{\text{E}}K^{\text{p}},1}$ are used to include the constraint (4.4) on the maximum outflow of the number n^{E} of exits in the network:

$$M_4^{\text{ineq}} = \begin{bmatrix} \ddots & & 0 \\ & \begin{bmatrix} M^{\text{I},4} & 0 \end{bmatrix} & \\ 0 & & \ddots \end{bmatrix} , \tag{4.68}$$

and the vector V_4^{ineq} is given as:

$$V_4^{\text{ineq}} = \begin{bmatrix} \dfrac{q_1^{\text{out,max}}(k_0^{\text{c}})}{q_1^{\text{sat}}} \\ \vdots \\ \dfrac{q_{n^{\text{E}}}^{\text{out,max}}(k_0^{\text{c}})}{q_{n^{\text{E}}}^{\text{sat}}} \\ \vdots \\ \dfrac{q_1^{\text{out,max}}(k_0^{\text{c}}+K^{\text{p}})}{q_1^{\text{sat}}} \\ \vdots \\ \dfrac{q_{n^{\text{E}}}^{\text{out,max}}(k_0^{\text{c}}+K^{\text{p}})}{q_{n^{\text{E}}}^{\text{sat}}} \end{bmatrix} , \tag{4.69}$$

where the matrix $M^4 \in \mathbb{R}^{n^{\text{E}},n^{\text{in,tot}}-n^{\text{states}}}$ is given as:

$$M^4 = \begin{bmatrix} \ddots & & 0 \\ & \begin{bmatrix} M^{\text{I},4} & 0 \end{bmatrix} & \\ 0 & & \ddots \end{bmatrix} , \tag{4.70}$$

with the matrix $M^{\mathrm{I},4}$ a zero matrix except for the diagonal elements that are related to exits which are set to 1.

The fifth and sixth matrices $M_5^{\mathrm{ineq}} \in \mathbb{R}^{n^{\mathrm{L}}+n^{\mathrm{O}},n^{\mathrm{in,tot}}}$ and $M_6^{\mathrm{ineq}} \in \mathbb{R}^{n^{\mathrm{L}}+n^{\mathrm{O}},n^{\mathrm{in,tot}}}$ and vectors $V_5^{\mathrm{ineq}} \in \mathbb{R}^{n^{\mathrm{L}}+n^{\mathrm{O}},1}$ and $V_6^{\mathrm{ineq}} \in \mathbb{R}^{n^{\mathrm{L}}+n^{\mathrm{O}},1}$ are used to limit the control signals according to (4.10) and (4.11) and are given as:

$$M_5^{\mathrm{ineq}} = \begin{bmatrix} \ddots & & 0 \\ & \begin{bmatrix} I & 0 \end{bmatrix} & \\ 0 & & \ddots \end{bmatrix}, \tag{4.71}$$

$$V_5^{\mathrm{ineq}} = 1\,, \tag{4.72}$$

$$M_6^{\mathrm{ineq}} = -M_5^{\mathrm{ineq}}\,, \tag{4.73}$$

$$V_6^{\mathrm{ineq}} = 0\,. \tag{4.74}$$

The matrix $M_7^{\mathrm{ineq}} \in \mathbb{R}^{n^{\mathrm{con}}K^{\mathrm{p}},n^{\mathrm{in,tot}}}$ and vector $V_7^{\mathrm{ineq}} \in \mathbb{R}^{n^{\mathrm{con}}K^{\mathrm{p}},1}$ take care of the conflicts (4.12) and are given as:

$$M_7^{\mathrm{ineq}} = \begin{bmatrix} \ddots & & 0 \\ & \begin{bmatrix} \bar{M}^{\mathrm{conflict}} & 0 \end{bmatrix} & \\ 0 & & \ddots \end{bmatrix}, \tag{4.75}$$

$$V_7^{\mathrm{ineq}} = 1\,, \tag{4.76}$$

with $\bar{M}^{\mathrm{conflict}}$ a matrix in which every row represents a conflict so that element $\bar{M}_{i,j}^{\mathrm{conflict}} = 1$ for every link j in the set I_i^{conflict} of conflict i and all the other entries are set to 0.

Chapter 5

Efficient Joint Optimization of Routing and Intersection Flows using the Link Transmission Model

This chapter extends the MPC strategy proposed in the previous chapter to jointly optimize the flows and the routing decisions in order to improve the urban network throughput. This chapter is based on the following paper that is currently under review:

G.S. van de Weg, E.-S. Smits, H. Taale, A. Hegyi, B. De Schutter, and S.P. Hoogendoorn, Efficient Joint Optimization of Routing and Intersection Flows using the Link Transmission Model. *Transportation Research Part C*, submitted 2017-03-25.

Abstract

One of the challenging problems of urban traffic control is the interaction between the chosen traffic light settings and the route choice of road users. This interaction causes that urban traffic control strategies optimized based on the real-time traffic states and historical data may get out of date and become less efficient over time. The reason for this is that people get acquainted with the travel times over their possible routes caused by the signal controllers and select new routes over time. One of the solutions to this problem is to explicitly control the routing decisions – e.g. using in-car navigation devices or route information signs – and jointly optimizing the traffic light signal controllers and the routing decisions. The design of such a control strategy is difficult because it consists of a large number of decision variables and requires a predictive control action that is computationally hard. In this paper, an efficient optimization algorithm is proposed for the joint optimization of traffic flows at intersections and en-route routing decisions assuming a 100% compliance rate in a model predictive control

framework. The algorithm is of the sequential linear programming type and uses an analytic procedure to approximate a linearization of the model around an operation point instead of a numerical linearization approach. Simulations using several optimization algorithms show that the proposed approach yields a better trade-off between computation time used and throughput improvements. Also, the simulations indicate the added value of the analytic linearization approach, and the use of the sequential linear programming algorithm.

5.1 Introduction

Urban road traffic network control using traffic lights influences the route-choice of individual road users [Taale and van Zuylen, 2001]. The reason for this is that traffic lights influence the travel times on road sections and therewith the total travel times of different routes between an origin and a destination. The implication of this effect is that an urban traffic control strategy, that is designed based on knowledge of current and historical flows, may get out of date and become less efficient over time due to the changed route choice of road users when they get acquainted with the traffic dynamics. The resulting equilibrium in route choice that appears in this way is commonly referred to as the user equilibrium [Wardrop and Whitehead, 1952]. The travel time on every used route from an origin to a destination is identical in the user equilibrium, and it is smaller than the travel time of the unused routes.

The efficiency loss of the signal controllers due to route choice may even occur quicker in the near future due to the proliferation of in-car technologies, such as, GPS navigation devices with which more and more vehicles are equipped. An even stronger effect may be expected when automated vehicles that automatically navigate from origin to destination become widely available. Such systems may choose and adopt the best route for the individual road user based on knowledge of the current traffic situation, possibly combined with a prediction of the evolution of the traffic state over time. This may cause an even faster degradation of the overall network performance. Hence, an urban traffic control strategy has to take into account the impact of its control action onto the route choice, potentially leading to a more efficient user equilibrium or it has to explicitly control the route choice behavior so that a system optimum can be achieved.

In general, system optimality may imply that alternative (used) routes between a given origin-destination pair have different travel times. This implies that some vehicles may have shorter travel times compared to others so that the average travel time experienced by all the road users is optimal. Currently most drivers minimize their individual cost by choosing the route with the lowest travel time (if known), but in the future incentives may be given by monetary tolls or rewards for drivers for choosing the route that leads to the system optimum [Pigou, 2013]. In such systems the drivers will still minimize their individual costs, but now for the generalized (combined) monetary and travel time

costs. In a dynamic setting, pricing is discussed in the bottleneck model by Vickrey [1969]. A more operational incentive can be tradable driving rights [Xiao et al., 2013].

A common approach to optimize the network performance is predictive control. An advantage of a predictive control action is that it allows to account for the future impact of control actions. This is useful in a traffic network, since, changing the flows or route choices at one intersection affects other intersections at a later time instant. One of the major challenges of predictive control is that it may lead to a computationally complex optimization problem. This is a challenge because the computation time required to solve the optimization problem must be smaller than the real-time controller sampling time.

Considering the desired aspects of traffic control, the aim of this paper is the design of a control strategy that:

- optimizes the throughput of an urban road traffic network over a time horizon
- controls the average traffic flows at intersections which may be realized by traffic lights and the route choice of the traffic
- requires limited computation time, more specifically, the computation time has to be less than the real-time controller sampling time which is in the range of (several) minutes

Before detailing the research approach and contributions, first Section 5.1.1 discusses approaches to the combined dynamic traffic assignment and signal control problem. After that, Section 5.1.2 details a specific sub-set of approaches, namely model-based optimization approaches.

5.1.1 Approaches to the combined dynamic traffic assignment and signal control problem

The problem of accounting for the impact of traffic signal control on route choice has drawn research attention for several decades. Already in 1974, Allsop [1974] discussed the interaction effect of signal control and route choice. Taale and van Zuylen [2001] presented an overview of the literature on the combined traffic assignment and control problem. The approaches to solve the combined traffic assignment and control problem can be divided into three categories, namely, 1) iterative procedures, 2) global optimization approaches, and 3) game-theoretic approaches that intend to solve the global optimization problem [Smith, 1985, Taale and van Zuylen, 2001].

In iterative approaches, the traffic signals settings are optimized for a given demand pattern. Next, the route choice of drivers is updated for the optimized signal settings, leading to a new demand pattern, and the process is repeated until convergence [Allsop and Charlesworth, 1977, Akçelik and Maher, 1977]. A comparison of several studies

that focused on the convergence properties of the iterative approach by Taale and van Zuylen [2001] indicated that a challenge of iterative approaches may be that they are not guaranteed to converge to a stable optimum despite the potential computation time gain.

Global optimization approaches address this issue by jointly optimizing the route choice and traffic signal settings [Taale and van Zuylen, 2001, Chen and Hsueh, 1997]. Due to the necessity to predict the impact of the traffic signals on the link travel times, a traffic assignment model is used. Hence, most of the combined traffic assignment and control approaches are of the model predictive control type. Recent developments in the area of model-based optimization approaches are discussed in the next section.

The first game-theoretic approaches to solve the optimization problem were presented in the 1980's [Fisk, 1984]. The idea is that road users and traffic managers can be modeled as different decision makers that have different objectives [Gartner et al., 1980, Taale and van Zuylen, 2001]. Chen [1998] proposed a dynamic modeling framework where control strategies and assignment can be combined. The advantage of using game theory is that it does not require the explicit use of the evaluation of computationally expensive prediction models while still being able to realize similar performance according to Taale and van Zuylen [2001].

5.1.2 Model-based optimization approaches

A common approach to predict and optimize the (future) impact of the control action onto the overall network performance is model-based optimization, commonly known as model predictive control (MPC). Several researchers have proposed model-based optimization strategies for the combined dynamic traffic assignment and control problem.

Taale and Hoogendoorn [2013] proposed a framework for real-time integrated and anticipatory traffic management that is somewhere in between iterative and global optimization approaches. The framework is similar to iterative approaches, but when optimizing the signal settings, the impact of the control settings on the route choice behavior is explicitly considered using a traffic flow model. In this way, better convergence is expected. The computation time of the iterative procedure may still be high. Abdul Aziz and Ukkusuri [2012] used the cell transmission model (CTM) as a prediction model in an MPC framework for optimization of the signal settings assuming system-optimal route choice behavior. The authors optimize the phase selection using a mixed-integer linear programming problem (MILP) but they do not optimize the route choice. Challenges of this approach are that the computation time is high due to the use of the MILP and that the linear formulation leads to violation of the first-in-first-out (FIFO) principle for the different destination-oriented flows in the cells.

Le et al. [2013] proposed an optimization approach based on a multi-class variant of the CTM. This means that it is similar to the CTM, except that the shock wave speed

of spillback is not modeled. The authors aggregate the traffic dynamics to (several) tens of seconds so that the discontinuous nature of the signal settings does not have to be considered. In this way, the authors are able to formulate a quadratic programming problem to optimize the flows at the intersections and the destination-oriented turn rates. Due to the use of (a variant of) the CTM in combination with a quadratic optimization problem, the approach may – similarly as in Abdul Aziz and Ukkusuri [2012] – violate the FIFO principle. Hence, the approach will only provide correct results when applied to networks with a single destination. Another weakness of the model used is that it does not reproduce the shock wave speed of spillback, leading to a performance degradation in oversaturated traffic conditions [van de Weg et al., 2016].

Li et al. [2015] optimize the route guidance and traffic signal settings using a space-phase-time hyper network. The authors decompose the problem into two sub-problems with different properties. Hence, the traffic signal optimization problem is optimized based on a phase-time network considering aggregated dynamics. The route guidance for individual vehicles is solved based on the link travel times. This procedure is repeated until convergence. An advantage of this work is the level of detail considered – i.e., the explicit inclusion of signal timings, and control of route decisions of individual road users. However, this also leads to a very complex model that runs with a resolution of 1 second.

This brief overview shows that there exist different model-based optimization approaches, each having its own advantages and challenges. However, to the best knowledge of the authors, an approach that can optimize the throughput in all traffic regimes – i.e. the undersaturated, saturated, and oversaturated regimes – using a computationally efficient optimization procedure does not exist yet. In this paper, a link is in the undersaturated regime when the queue can fully clear when given green, it is in the saturated regime when the queue does not clear when given green and neither spills back to upstream intersections, and it is in the oversaturated regime when the queue that spills back and cause blocking of upstream intersections.

5.1.3 Research approach and contributions

The aim of this paper is to develop an algorithm for the joint optimization of intersection flows and route decisions that is computationally efficient and is able to improve the network throughput in all traffic regimes. In order to reach this goal, the following simplifications are made. First of all, a 100% compliance rate to the optimized route choice is assumed. In practice, this may be realized using, for instance, (monetary) incentives as discussed above. Second, the traffic dynamics are aggregated to several (tens of) seconds. Due to this, the intersection flows are continuous and signal timings are not explicitly considered so that the objective function of the optimization problem is differentiable everywhere with respect to the control variables. Hence, gradient-based solvers can be used, which are generally faster when compared to gradient-free

solvers when the problem size is not too large. Third, it is assumed that the origin-destination demands are known and that no noise or uncertainties affect the system. Hence, in practice a module may be needed to predict the demand. It must be noted that both the quality of the optimization and the demand predictions may influence the controller performance. Because this paper aims at studying the quality of the optimization, simulations are carried out in a controlled environment that are primarily focused on investigating the impact of the controller on the throughput and computation time.

In the light of these design considerations, this paper proposes an efficient model predictive control strategy based on the link transmission model (LTM) for the combined optimization of intersection flows and route choice. The LTM is used because it is a more computationally efficient model when compared to segment-based models, such as the CTM, as shown in van de Weg et al. [2016]. Due to the non-linear nature of the optimization problem, an efficient optimization algorithm of the sequential linear programming type is proposed [Marcotte and Dussault, 1989]. The idea behind the algorithm is that first the non-linear model is used to predict the traffic state trajectories for a candidate control signal. Based on that prediction, an analytic approximation of the model linearization is determined which is used to formulate a linear optimization problem which is solved, giving a new control signal. Next, the candidate control signal and the optimized control signal are used as a search direction in a line-search optimization algorithm giving a new candidate control signal. Finally, it is tested whether the control signal found satisfies the stopping criteria. If not, the process is repeated. Compared to the research discussed above, advantages are the use of the LTM, the improved, analytic approximation of the model linearization, and the use of the line-search algorithm.

The contributions of this paper are:

- Design of an MPC strategy using the LTM for optimization of both intersection flows and routing decisions (Section 5.3)
- Design of an efficient optimization algorithm based on analytic approximation of the model linearization (Section 5.3)
- Formulation of an analytic approximation of the model linearization around an operating point (Section 5.3.3)
- Evaluation of the approach in terms of the trade-off between realized throughput and CPU time used by comparing with different variants of the SLP algorithm and a numerical linearization based optimization algorithm within a simulation environment (Section 5.4)

The paper is structured as follows. First, Section 5.2 details the traffic flow prediction model that is used. Second, Section 5.3 introduces the optimization algorithm and the

analytic linearization of the model. Section 5.4 evaluates the approach using simulations. Section 5.5 concludes the paper. 5.B provides an overview of the variables used in the paper.

5.2 Description of traffic flow dynamics

This section details the macroscopic traffic flow model used to describe the evolution of the traffic in an urban road network. The link dynamics are modeled using the link transmission model of Yperman [2007], the node dynamics are modeled using the directed capacity proportional node model presented in Smits et al. [2015], origins are modeled as vertical queues, and destinations are modeled as sinks. An overview of the network elements used in the paper is shown in Figure 5.1, an overview of the variables used is given in 5.B. The LTM is chosen for two reasons. First of all, the LTM is a computationally efficient traffic flow model that is capable of reproducing the most important link dynamics – it describes both forward moving free-flow waves, and backward propagating queue tails and heads. The latter is an important advantage when compared to other approaches that model the link dynamics using segments, causing an increase in computational overhead. Secondly, as will be shown in Section 5.3.3, an analytic procedure to linearize the LTM and node model around an operating point can be derived so that no numerical schemes have to be used to determine the gradient, which results in a computation time gain. The LTM is a first-order traffic flow model. This model type ignores higher-order dynamics, such as acceleration behavior, so that it is more computationally efficient. Nevertheless, despite ignoring these higher-order effects, it is capable of reproducing the most important characteristics of the traffic dynamics required for control as discussed above.

The main traffic states that are to be updated by the LTM are the cumulative inflows $N_l^{\text{in}}(k)$ (veh) and outflows $N_l^{\text{out}}(k)$ (veh) of all the links $l \in \mathcal{I}^{\text{L}}$ (-) with $\mathcal{I}^{\text{L}}$ the set of all the links, and the cumulative inflows $N_o^{\text{in}}(k)$ (veh) and outflows $N_o^{\text{out}}(k)$ (veh) of all the origins $o \in \mathcal{I}^{\text{O}}$ (-) with $\mathcal{I}^{\text{O}}$ the set of all the origins according to the following equations:

$$N_l^{\text{in}}(k+1) = N_l^{\text{in}}(k) + q_l^{\text{in}}(k)T\,, \tag{5.1}$$

$$N_l^{\text{out}}(k+1) = N_l^{\text{out}}(k) + q_l^{\text{out}}(k)T\,, \tag{5.2}$$

$$N_o^{\text{in}}(k+1) = N_o^{\text{in}}(k) + q_o^{\text{in}}(k)T\,, \tag{5.3}$$

$$N_o^{\text{out}}(k+1) = N_o^{\text{out}}(k) + q_o^{\text{out}}(k)T\,, \tag{5.4}$$

With the flows $q_l^{\text{in}}(k)$ (veh/h) and $q_l^{\text{out}}(k)$ (veh/h) the link inflows and outflows at time step k, and the flows $q_o^{\text{in}}(k)$ (veh/h) and $q_o^{\text{out}}(k)$ (veh/h) the origin inflows and outflows at time step k. The time T (h) is the model sampling time. In order to update the traffic states, several parameters have to be known. These are, the time t_l^{free} (h) it takes to travel through the link in free-flow, the time t_l^{shock} (h) it takes a backward propagating

shock wave to travel through the link, the saturation rate q_l^{sat} (veh/h), and the maximum number of vehicles $N_l^{\max}$ (veh) that fits in the link.

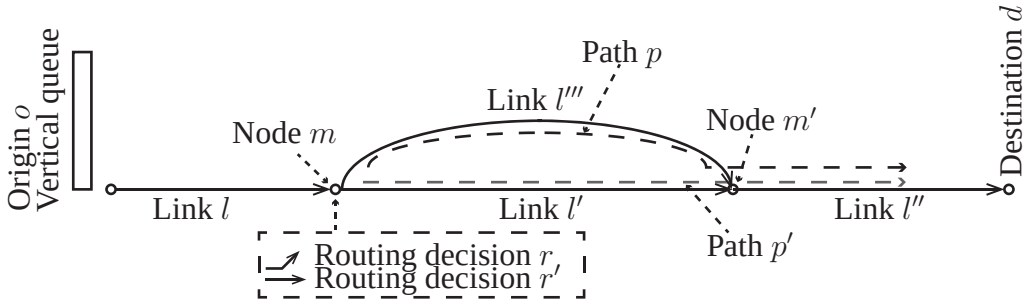


Figure 5.1: Overview of network elements and indexes used in the paper.

Additionally, every link and origin in the network can be connected to several destinations $d \in \mathcal{I}^{\text{D}}$ (-) with $\mathcal{I}^{\text{D}}$ the set of all destinations in the network. On every link the destination-oriented cumulative inflows $N_{l,d}^{\text{in}}(k)$ (veh) and outflows $N_{l,d}^{\text{out}}(k)$ (veh), and on every origin the destination-oriented cumulative inflows $N_{o,d}^{\text{in}}(k)$ (veh) and outflows $N_{o,d}^{\text{out}}(k)$ (veh) are updated according to the following equations:

$$N_{l,d}^{\text{in}}(k+1) = N_{l,d}^{\text{in}}(k) + q_{l,d}^{\text{in}}(k)T\,, \tag{5.5}$$
$$N_{l,d}^{\text{out}}(k+1) = N_{l,d}^{\text{out}}(k) + q_{l,d}^{\text{out}}(k)T\,, \tag{5.6}$$
$$N_{o,d}^{\text{in}}(k+1) = N_{o,d}^{\text{in}}(k) + q_{o,d}^{\text{in}}(k)T\,, \tag{5.7}$$
$$N_{o,d}^{\text{out}}(k+1) = N_{o,d}^{\text{out}}(k) + q_{o,d}^{\text{out}}(k)T\,, \tag{5.8}$$

where the flows $q_{l,d}^{\text{in}}(k)$ (veh/h) and $q_{l,d}^{\text{out}}(k)$ (veh/h) are the inflows and outflows of traffic with destination d on link l, and the flows $q_{l,d}^{\text{in}}(k)$ (veh/h) and $q_{l,d}^{\text{out}}(k)$ (veh/h) are the inflows and outflows of traffic with destination d on origin o

Hence, the main task of the model is computing at every time step the inflows and outflows. The model can be described in several steps, which are executed when updating the traffic flow dynamics from one time step to another. In this brief overview we will detail the different steps first without using any equations:

1. The first step is to update the maximum cumulative link outflow. This is done by taking the minimum of the cumulative free-flow outflow, the outflow under saturated conditions, and the capacity of a downstream bottleneck as will be detailed in Section 5.2.1.

2. Next, the travel time that corresponds to the computed cumulative link outflow is determined as will be detailed in Section 5.2.2.

3. Using the travel time in the link, the maximum cumulative destination-oriented outflows on the link can be determined as will be detailed in Section 5.2.3.

4. For every link, the maximum inflow is computed as will be detailed in Section 5.2.4.

5. Based on the maximum (destination-oriented) outflow and the maximum inflows, the turn fractions at the nodes are computed. Next, a node model is applied to compute reduction factors with which the maximum link outflows are multiplied in order to reduce the link outflows in the event of capacity conflicts due to spill back as will be detailed in Section 5.2.6.

6. Finally, using the reduction factors the cumulative inflows and outflows can be determined as will be detailed in Section 5.2.7.

5.2.1 Updating the maximum cumulative link outflow

The first step is to determine the maximum sending flow $q_l^{\text{out,max}}(k)$ (veh/h) of a link. The sending flow is determined by various factors that depend on the traffic condition in the link. When there is no queue in the link, the maximum sending flow is equal to the free-flow outflow. When there is a queue, the maximum sending flow is equal to the saturation rate q_l^{sat} (veh/h), multiplied with the effective green fraction $b_l(k)$ (-). The fraction $b_l(k)$ is defined as the realized link outflow divided by the saturation rate. In the case that a link is at the exit of the network, the outflow may be limited by a gating policy or by spillback from a downstream road that is not modeled, resulting in a maximum outflow $q_l^{\text{bn}}(k)$ (veh/h). Thus, the cumulative maximum outflow $N_l^{\text{out,max}}(k+1)$ (veh) can be updated as follows

$$\begin{aligned} N_l^{\text{out,max}}(k+1) = \min \Big[& N_l^{\text{out}}(k) + q_l^{\text{sat}} b_l(k) T, \dots \\ & \gamma_l^{\text{free}} N_l^{\text{in}}(k+2-k_l^{\text{free}}) + (1-\gamma_l^{\text{free}}) N_l^{\text{in}}(k+1-k_l^{\text{free}}), \dots \\ & N_l^{\text{out}}(k) + q_l^{\text{bn}}(k) T \Big] . \end{aligned} \tag{5.9}$$

Here, k_l^{free} (-) is the number of time steps required to travel through the link in free flow, which can be computed from the free-flow travel time t_l^{free} (h) as $k_l^{\text{free}} = \lceil t_l^{\text{free}}/T \rceil$, and the fraction γ_l^{free} (-) is the fraction of time step T that k_l^{free} exceeds the free-flow travel time: $\gamma_l^{\text{free}} = k_l^{\text{free}} - t_l^{\text{free}}/T$. The mathematical operator $\lceil \cdot \rceil$ means rounding to the nearest integer that is equal to or larger than the argument of the function. It must hold that $k_l^{\text{free}} \geq 2$ in order to satisfy CFL conditions.

The maximum sending flow $q_l^{\text{out,max}}(k)$ (veh/h) from a link is then given as the maximum number of vehicles $N_l^{\text{out,max}}(k+1) - N_l^{\text{out}}(k)$ (veh) that can exit the link divided by the time step:

$$q_l^{\text{out,max}}(k) = \frac{N_l^{\text{out,max}}(k+1) - N_l^{\text{out}}(k)}{T} . \tag{5.10}$$

5.2.2 Link travel time

The travel time $t_l^{\mathrm{tr}}(t)$ on a link at time t is given as the horizontal distance in the time-cumulative outflow diagram between the cumulative inflow and outflow curve as illustrated in Figure 5.2. The travel time is used to update the destination-oriented flows as will be explained in the next subsection. According to Yperman [2007], the travel time can be derived from the cumulative curves using the inverse of the cumulative curve as illustrated in Figure 5.2:

$$t_l^{\mathrm{tr}}(t) = t - N_l^{\mathrm{in}^{-1}}(N_l^{\mathrm{out}}(t))\,. \tag{5.11}$$

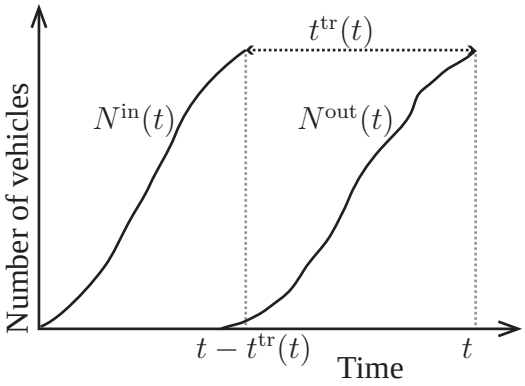


Figure 5.2: The relation between the cumulative inflow, the cumulative outflow, and the travel time at time t.

Opposed to the above equations which are in continuous time, the traffic dynamics in this paper are described in discrete time. Denote with the time $t_l^{\mathrm{tr}}(k)$ (h), the travel time that vehicles exiting the link at time step k have experienced. In such a case, the cumulative outflow and inflow are related via the following equation:

$$N_l^{\mathrm{out}}(k) = \gamma_l^{\mathrm{tr}}(k) N_l^{\mathrm{in}}(k + 1 - k^{\mathrm{tr}}(k)) + (1 - \gamma_l^{\mathrm{tr}}(k)) N^{\mathrm{in}}(k - k_l^{\mathrm{tr}}(k))\,, \tag{5.12}$$

with $k_l^{\mathrm{tr}}(k)$ (-) the number of time steps it takes to travel through the link: $k_l^{\mathrm{tr}}(k) = \lceil t_l^{\mathrm{tr}}(k)/T \rceil$ and $\gamma_l^{\mathrm{tr}}(k) = k_l^{\mathrm{tr}}(k) - t_l^{\mathrm{tr}}(k)/T$ the fraction of the sampling time T that this travel time exceeds the travel time $t_l^{\mathrm{tr}}(k)$ (h). Figure 5.3 provides a graphical representation of the approach. In this hypothetical example where the travel time $t_l^{\mathrm{tr}}(k)$ is 47 seconds and the sampling time is 10 seconds, it can be observed that the cumulative outflow at time step k is a linear combination of the cumulative outflows at time steps $k - 4$ and $k - 5$. Equation (5.12) provides a procedure to compute the cumulative outflow when the link travel time is known. When the cumulative outflow is known at time step k, the equation can be used to determine the travel time as well. The interested reader is referred to Long et al. [2011] for a detailed analysis of the link travel time based on cumulative flows.

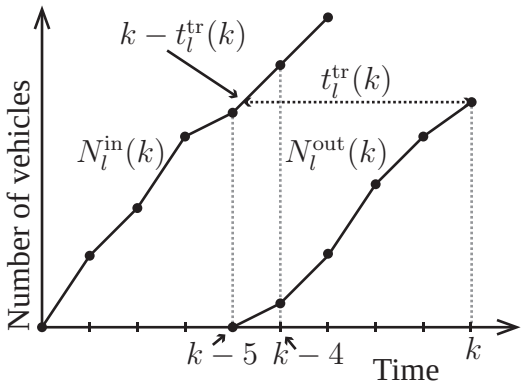


Figure 5.3: The relation between the cumulative inflow, the cumulative outflow, and the travel time at time step k for a hypothetical example where the current travel time is $t_l^{\mathrm{tr}}(k)$ is 47 seconds and the sampling time is 10 seconds. It can be observed that the cumulative outflow at time step k is a linear combination of the inflow at time steps $k-4$ and $k-5$.

5.2.3 Updating destination-oriented outflows

Based on the travel time $t_i^{\mathrm{tr}}(k)$ in a link, the destination-oriented outflows of a link can be updated. Denote with $N_{l,d}^{\mathrm{in}}(k)$ (veh) and $N_{l,d}^{\mathrm{out,max}}(k)$ (veh) the cumulative inflow and maximum outflow of traffic to destination d on link l. Now, the outflow is related to the inflow according to the following equation:

$$\begin{aligned} &N_{l,d}^{\mathrm{out,max}}(k+1) = \dots \\ &\gamma_l^{\mathrm{tr}}(k+1) N_{l,d}^{\mathrm{in}}(k+1-k_l^{\mathrm{tr}}(k+1)) + (1-\gamma_l^{\mathrm{tr}}(k+1)) N_{l,d}^{\mathrm{in}}(k-k_l^{\mathrm{tr}}(k+1))\,. \end{aligned} \tag{5.13}$$

The maximum destination-oriented outflow $q_{l,d}^{\mathrm{out,max}}(k)$ (veh/h) is then given as:

$$q_{l,d}^{\mathrm{out,max}}(k) = \frac{N_{l,d}^{\mathrm{out,max}}(k+1) - N_{l,d}^{\mathrm{out}}(k)}{T}\,. \tag{5.14}$$

Applying this equation ensures that the FIFO principle with respect to the different destination-oriented flows on a link is satisfied.

5.2.4 Updating the maximum cumulative link inflow

The maximum receiving flow $q_l^{\mathrm{in,sp}}(k)$ (veh/h) explicitly considers the impact of the shock wave dynamics in the LTM. The cumulative inflow is restricted by the maximum number of vehicles $N_i^{\mathrm{in,max}}(k)$ (veh) that can enter the link when the link is full:

$$\begin{aligned} &N_l^{\mathrm{in,max}}(k+1) = \dots \\ &\gamma_l^{\mathrm{shock}} N_l^{\mathrm{out}}(k+2-k_l^{\mathrm{shock}}) + (1-\gamma_l^{\mathrm{shock}}) N_l^{\mathrm{out}}(k+1-k_l^{\mathrm{shock}}) + N_l^{\mathrm{max}}\,, \end{aligned} \tag{5.15}$$

where the number N_l^{max} (veh) represents the maximum number of vehicles that fits in link l, and the number of time steps k_l^{shock} (-) represents the number of time steps

it takes a shock wave to travel through the link. The fraction γ_l^{shock} (-) is given by: $\gamma_l^{\text{shock}} = t_l^{\text{shock}}/T - k_l^{\text{shock}}$ with t_l^{shock} (h) being the shock wave travel time. Note that it must hold that $k_l^{\text{shock}} \geq 2$ to satisfy the CFL conditions.

Now, the maximum receiving flow $q_l^{\text{in,sp}}(k)$ (veh/h) is given as the minimum of the saturation rate and the maximum link inflow:

$$q_l^{\text{in,sp}}(k) = \min\left[\frac{N_l^{\text{in,max}}(k+1) - N_l^{\text{in}}(k)}{T}, q_l^{\text{sat}}\right]. \tag{5.16}$$

5.2.5 Updating the origin inflows and outflows

Origins are modeled as vertical queues and are described in this section. Recall that it is assumed that the destination-oriented demands $q_{o,d}^{\text{demand}}(k)$ (veh/h) are given. The first step in updating origins is computing the destination-oriented inflows $q_{o,d}^{\text{in}}(k)$ (veh/h):

$$q_{o,d}^{\text{in}}(k) = q_{o,d}^{\text{demand}}(k)\,. \tag{5.17}$$

The total origin inflow $q_o^{\text{in}}(k)$ (veh/h) is then given as:

$$q_o^{\text{in}}(k) = \sum_{d \in \mathcal{I}_o^{\text{D}}} q_{o,d}^{\text{in}}(k)\,. \tag{5.18}$$

The origin outflow $q_o^{\text{out}}(k)$ is given as the minimum of the cumulative origin inflow $N_o^{\text{in}}(k+1)$ (veh) at time step k, the maximum link inflow $N_l^{\text{in,max}}(k+1$ (veh/) of the downstream link l, and the origin capacity q_o^{cap} (veh/h):

$$q_o^{\text{out}}(k) = \frac{\min\left(N_o^{\text{in}}(k+1), N_l^{\text{in,max}}(k+1), N_o^{\text{out}}(k) + q_o^{\text{cap}}\right) - N_o^{\text{out}}(k)}{T}. \tag{5.19}$$

Using the origin outflow, the origin travel time $t_o^{\text{tr}}(k+1)$ (h) of the vehicles exiting the origin at time step $k+1$ can be derived, similarly as in Section 5.2.2. Using this travel time, the destination-oriented outflows $q_{o,d}^{\text{out}}(k)$ (veh/h) can be computed as follows:

$$\begin{aligned} q_{o,d}^{\text{out}}(k) = &\frac{\gamma_o^{\text{tr}}(k+1) N_{o,d}^{\text{in}}(k+1-k_o^{\text{tr}}(k+1))}{T} + \dots \\ &\frac{(1-\gamma_o^{\text{tr}}(k+1)) N_{o,d}^{\text{in}}(k-k_o^{\text{tr}}(k+1)) - N_{o,d}^{\text{out}}(k)}{T}. \end{aligned} \tag{5.20}$$

5.2.6 The node model

Node models determine the flow over nodes given boundary conditions provided by the adjacent incoming and outgoing links. Tampère et al. [2011] provide a basic set of requirements for node models. These requirements assure that the node model results are consistent with basic behavior of drivers. Currently, four models are known that

satisfy these requirements according to Smits et al. [2015]. This study adopts the solution method for the directed capacity proportional node model presented in Smits et al. [2015]. This node model was first introduced by Flötteröd and Rohde [2011] and Tampère et al. [2011].

The node model is used to connect links and to distribute the flow from the incoming links $l \in \mathcal{I}_m^{\mathrm{L,us}}$ to the outgoing links $l' \in \mathcal{I}_m^{\mathrm{L,ds}}$ of the node m (-). It is also used to distribute spillback from the outgoing links to the incoming links over the node. The main task of the node model is to determine the reduction factors $\beta_l(k)$ (-) of the links $l \in \mathcal{I}_m^{\mathrm{L,us}}$ upstream of the node m, such that it holds that $q_{l'}^{\mathrm{in}}(k) \leq q_{l'}^{\mathrm{in,sp}}(k)$. Node models require as input the turn fractions $\eta_{l,l'}(k)$ (-) of traffic from upstream links to downstream links, the maximum link outflows $q_l^{\mathrm{out,max}}(k)$ (veh/h) – i.e. the demand – of the incoming links, the maximum link inflows $q_{l'}^{\mathrm{in,sp}}(k)$ (veh/h) – i.e. the supply – of outgoing links, and the saturation rate q_l^{sat} (veh/h) of all the links connected to the node m.

The turn fraction $\eta_{l,l'}(k)$ of traffic on link l towards link l' is defined as the total sum of demand on link l towards link l' divided by the total demand of link l:

$$\eta_{l,l'}(k) = \frac{\sum_{d \in (\mathcal{I}_l^{\mathrm{D}} \cap \mathcal{I}_{l'}^{\mathrm{D}})} \bar{u}_{l,l',d}^{\mathrm{D}}(k) q_{l,d}^{\mathrm{out,max}}(k)}{q_l^{\mathrm{out,max}}(k)} \,. \tag{5.21}$$

Here, the fraction $\bar{u}_{l,l',d}^{\mathrm{D}}(k)$ (-) is a control variable that controls the fraction of traffic oriented to destination d on link l that will travel via downstream link l'. It is assumed that the turn fractions $\eta_{l,l'}(k)$ of traffic from upstream links to downstream links remain constant during the current time step.

In the model, the directed capacity – i.e., the capacity of the corresponding incoming link multiplied with the turn fraction –, determines if traffic will spill back towards an incoming link. The model ensures for example that in the case of spillback at a merge of a two-lane and one-lane road, the outflow of the two-lane is twice the outflow of the one-lane road. We refer to [Tampère et al., 2011, Flötteröd and Rohde, 2011, Smits et al., 2015] for details on the characteristics and underlying behaviour of this node model. The directed capacity node model can be summarized using the following function:

$$\beta_l(k) = f^{\mathrm{node}}\left(q_l^{\mathrm{out,max}}(k), q_l^{\mathrm{sat}}, q_{l'}^{\mathrm{in,sp}}(k), \eta_{l,l'}(k)\right) \,, \forall\, l \in \mathcal{I}_m^{\mathrm{L,us}} \,\&\, l' \in \mathcal{I}_m^{\mathrm{L,ds}} \,. \tag{5.22}$$

For the details of the mathematical formulation of this function the reader is referred to Algorithm 1 in [Smits et al., 2015].

5.2.7 Updating the link inflows and outflows

Now that the reduction factors $\beta_l(k)$ are known, the cumulative (destination-oriented) link inflows and outflows can be computed. First of all, the link outflows $q_l^{\mathrm{out}}(k)$

and destination-oriented outflows $q_{l,d}^{\mathrm{out}}(k)$ are computed by multiplying the maximum outflow $q_{l}^{\mathrm{out,max}}(k)$ and $q_{l,d}^{\mathrm{out,max}}(k)$ with the reduction factor:

$$q_{l}^{\mathrm{out}}(k) = \beta_l(k) q_{l}^{\mathrm{out,max}}(k)\,, \tag{5.23}$$
$$q_{l,d}^{\mathrm{out}}(k) = \beta_l(k) q_{l,d}^{\mathrm{out,max}}(k)\,. \tag{5.24}$$

After that, the destination-oriented link inflows $q_{l',d}^{\mathrm{in}}(k)$ (veh/h) are computed by summing up the destination-oriented outflows of upstream links multiplied with the en-route decision variable $\bar{u}_{l,l',d}^{\mathrm{D}}(k)$ – i.e., the fraction of the flow oriented to destination d on link l that travels via link l' – and adding the destination specific inflow of origins:

$$q_{l',d}^{\mathrm{in}}(k) = \sum_{l \in \mathcal{I}_{l'}^{\mathrm{L,us}}} \bar{u}_{l,l',d}^{\mathrm{D}}(k) q_{l,d}^{\mathrm{out}}(k) + \sum_{o \in \mathcal{I}_{l'}^{\mathrm{O,us}}} q_{o,d}^{\mathrm{out}}(k)\,. \tag{5.25}$$

Then, the total link inflow $q_{l'}^{\mathrm{in}}(k)$ is given as:

$$q_{l'}^{\mathrm{in}}(k) = \sum_{d \in \mathcal{I}_{l'}^{\mathrm{D}}} q_{l',d}^{\mathrm{in}}(k)\,. \tag{5.26}$$

5.3 The optimization algorithm

The objective of the optimization algorithm is minimizing the total time spent (TTS) Z^{TTS} (veh·h) by all the vehicles in the network including origins over a prediction horizon of K^{p} (-) steps subject to inequality constraints:

$$\begin{aligned} &\min_{U} Z^{\mathrm{TTS}}(U, D, X_0)\,, \\ &\text{s.t. } M^{\mathrm{ineq}} U \leq V^{\mathrm{ineq}}\,, \end{aligned} \tag{5.27}$$

Here, X_0 is the initial traffic state, the matrix D contains the disturbances, and the vector U contains the control signal. The vector U is defined as:

$$U = \begin{bmatrix} U(k) & U(k+1) & \cdots & U(k+K^{\mathrm{p}}-1) \end{bmatrix}^{\top}\,, \tag{5.28}$$

with

$$U(k) = \begin{bmatrix} b_1(k) & \cdots b_{n^{\mathrm{con}}}(k) & u_1^{\mathrm{D}}(k) & \cdots & u_{n^{\mathrm{decisions}}}^{\mathrm{D}}(k) \end{bmatrix}^{\top}\,, \tag{5.29}$$

with n^{con} (-) the number of controlled links, and $n^{\mathrm{decisions}}$ (-) the number of en-route decisions.

The routing decisions $u_r^{\mathrm{D}}(k)$ are mapped to the routing decisions $\bar{u}_{l,l',d}^{\mathrm{D}}(k)$ in the traffic flow model in the following way. First, denote with $\mathcal{I}_{l,d}^{\mathrm{L,RD}}$ (-) the ordered set of link indexes over which flows oriented to destination d on link l can be send. The number of links in this set equals $n^{\mathrm{RD}}+1$. Next, denote with $\mathcal{I}_{l,d}^{\mathrm{RD}}$ (-) the ordered set of indexes of the routing decision variables r on link l related to destination d. This set contains

n^{RD} variables. The sets are ordered so that the routing decision $r = \mathcal{I}_{l,d}^{\mathrm{RD}}(i)$ relates to link $l' = \mathcal{I}_{l,d}^{\mathrm{L,RD}}(i)$. Hence, the routing decision variables $u_r^{\mathrm{D}}(k)$ are mapped to the decision variables $\bar{u}_{l,l',d}^{\mathrm{D}}(k)$ in the following way:

$$\bar{u}_{l,l',d}^{\mathrm{D}} = u_r^{\mathrm{D}}(k)\,, \text{with } l' = \mathcal{I}_{l,d}^{\mathrm{L,RD}}(i)\,, r = \mathcal{I}_{l,d}^{\mathrm{RD}}(i)\,, \forall i \leq n^{\mathrm{RD}}\,, \tag{5.30}$$

$$\bar{u}_{l,l',d}^{\mathrm{D}} = 1 - \sum_{r \in \mathcal{I}_{l,d}^{\mathrm{RD}}} u_r^{\mathrm{D}}(k)\,, \text{with } l' = \mathcal{I}_{l,d}^{\mathrm{L,RD}}(n^{\mathrm{RD}}+1)\,. \tag{5.31}$$

In the case that there are only two downstream links l' and l'' in the set $\mathcal{I}_{l,d}^{\mathrm{L,RD}}$ this simplifies to:

$$\bar{u}_{l,l',d}^{\mathrm{D}} = u_r^{\mathrm{D}}(k)\,, \tag{5.32}$$

$$\bar{u}_{l,l'',d}^{\mathrm{D}} = 1 - u_r^{\mathrm{D}}(k)\,. \tag{5.33}$$

The matrix M^{ineq} and vector V^{ineq} are used to include the linear inequality constraints. The constraints are used to limit the green-fractions $b_l(k)$ between 0 and 1:

$$0 \leq b_l(k) \leq 1\,. \tag{5.34}$$

Additionally, it is ensured that the sum of the green-fractions of conflicting links is less than or equal to 1:

$$\sum_{l \in \mathcal{I}_c^{\mathrm{conflict}}} b_l(k) \leq 1\,. \tag{5.35}$$

with the set $\mathcal{I}_c^{\mathrm{conflict}}$ (-) containing the links that are in the conflict with index c (-). Third, the routing decisions $u_r^{\mathrm{D}}(k)$ are constrained between a minimum $u^{\mathrm{D,min}}$ (-) and a maximum $u^{\mathrm{D,max}}$:

$$u^{\mathrm{D,min}} \leq u_r^{\mathrm{D}}(k) \leq u^{\mathrm{D,max}}\,. \tag{5.36}$$

Fourth, it has to hold that the sum of routing decisions $u_r^{\mathrm{D}}(k)$ in a set $\mathcal{I}_{l,d}^{\mathrm{RD}}$ is smaller than or equal to 1:

$$0 \leq \sum_{r \in \mathcal{I}_{l,d}^{\mathrm{RD}}} u_r^{\mathrm{D}}(k) \leq 1\,. \tag{5.37}$$

This optimization problem is non-linear due to the non-linear traffic flow model that is used to predict the traffic state. The non-linearity is caused by updating the destination-oriented link outflows using the travel time, as detailed in Section 5.2.2 and Section 5.2.3, and by the node model. Due to these non-linearities, the computation time will grow very fast with increasing network size.

In order to reduce the computation time required by this algorithm, this section proposes an efficient optimization algorithm. The algorithm is of the Sequential Linear Programming (SLP) type [Marcotte and Dussault, 1989]. The computation time is mainly improved due the analytic procedure that is developed in this paper to approximate a linearization of the model around an operating point.

5.3.1 Overview of the SLP algorithm

Figure 5.1 provides an overview of the SLP algorithm. First, the algorithm is initialized by selecting an initial candidate control signal U_1 for which the model detailed in the previous section is run to get a prediction of the traffic state $X(U_1)$. After that, an iterative procedure is started at iteration $i = 1$ in which the following steps are carried out:

1. An effective control signal vector $\tilde{U}_i$ based on the predicted state $X(X_0, U_i, D)$ is determined.
2. Next, a linearization of the model around the point $\tilde{U}_i$ is approximated using an analytic procedure so that a linear optimization problem can be formulated. By solving the linear optimization problem (LP), the optimal control signal vector $\bar{U}_i^*$ is obtained.
3. After that, a line-search is carried out from the initial point $\tilde{U}_i$ in the direction of the optimized signal $\bar{U}_i$ which is a one-dimensional optimization problem. This line-search provides a new candidate control signal vector U_{i+1}.
4. The model is run again for this signal providing a new prediction of the traffic state $X(X_0, U_{i+1}, D)$ and a new value $Z^{\mathrm{TTS}}(U_{i+1})$ of the objective function.
5. The last step of every iteration is to check whether a stopping criterion is satisfied. If so, the signal U_{i+1} is the optimal signal U^* and is applied to the process. If the stopping criteria are not satisfied, the process is repeated for $i = i + 1$.

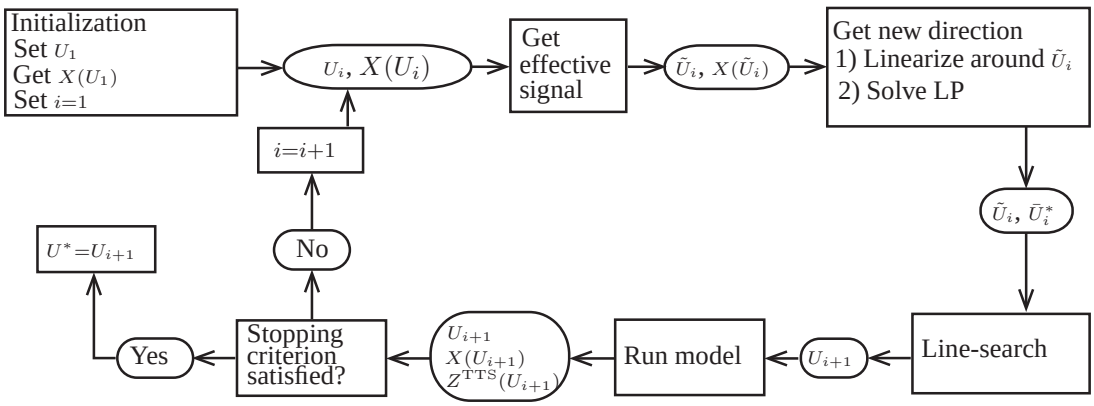


Figure 5.1: Overview of the SLP algorithm.

This section is structured as follows. First, Section 5.3.2 explains how the effective control signal is obtained. Next, Section 5.3.3 details the analytic procedure to approximate the linearization of the model, and Section 5.3.4 introduces the linear optimization problem that has to be solved. Section 5.3.5 details the line-search, and Section 5.3.6 presents the stopping criteria of the algorithm. Section 5.3.7 discusses the properties and limitations of the proposed algorithm.

5.3.2 The effective control signal

At every iteration of the SLP algorithm, the first step is to translate the control signal U_i into the effective control signal $\tilde{U}_i$. The effective control signal is defined as the fraction of green time that is used by the traffic. For instance, when setting the saturation rate to 2000 veh/h and applying a green fraction of 1.0 it may be the case that only a flow of 400 veh/h is realized. In that case, the effective green fraction would be 0.2 (-).

The effective control signal is obtained by replacing $U_i(k)$ with $\tilde{U}_i(k)$ in (5.28). The control signal $\tilde{U}_i(k)$ is defined as:

$$\tilde{U}(k) = \begin{bmatrix} b_1^{\text{eff}}(k) & \cdots b_{n^{\text{con}}}^{\text{eff}}(k) & u_1^{\text{D}}(k) & \cdots & u_{n^{\text{decisions}}}^{\text{D}}(k) \end{bmatrix}^\top , \tag{5.38}$$

where the fraction $b_l^{\text{eff}}(k)$ (-) is the effective fraction of green time used by the link. The fraction $b_l^{\text{eff}}(k)$ is obtained from the model output $X(X_0, U_i, D)$ by dividing the realized outflow of a link with the link saturation rate:

$$b_l^{\text{eff}}(k) = \frac{q_l^{\text{out}}(k)}{q_l^{\text{sat}}} . \tag{5.39}$$

5.3.3 Model linearization

The second step of the SLP algorithm is to approximate a linearization of the model around the signal $\tilde{U}_i(k)$. Note that the non-linearity in the model originates from different sources:

1. At every time step, the cumulative link inflow $N_l^{\text{in}}(k)$ is a function of the green fraction $b_{l'}(k)$ of upstream links, and of the en-route routing decisions $\bar{u}_{l',d}^{\text{D}}(k)$. This is a multiplicative effect, hence, it contains a non-linearity.

2. Updating the destination-oriented flows on the links and origins introduces a non-linearity because it requires to compute the travel time on the link and to allocate the link outflow to the different destinations proportional to the composition of the destination-oriented inflows at the time instant when they entered the link.

3. Finally, the solution procedure of the node model introduces non-linearities as well.

The most common way to linearize the model in the point $\tilde{U}_i(k)$ is numerical linearization, i.e., evaluating the objective function for changes in every element of the vector $\tilde{U}_i(k)$ so that the derivative w.r.t. every element can be computed. However, this requires a large number of function evaluations, resulting in a high computational complexity of the algorithm.

In order to reduce the computational complexity, an analytic approximation of the model linearization is proposed based on the linear MPC strategy proposed by van de Weg et al. [2016]. In that approach, a linear optimization problem can be formulated given that the turn fractions $\eta_{l',l}(k)$ are known so that the cumulative link inflow equation (5.25) can be simplified to:

$$N_l^{\text{in}}(k+1) = N_l^{\text{in}}(k) + \sum_{l' \in \mathcal{I}_l^{\text{L,us}}} \eta_{l',l}(k) q_{l'}^{\text{sat}} T^{\text{c}} b_{l'}^{\text{eff}}(k)\,. \tag{5.40}$$

Because the green fractions $b_l(k)$ are replaced with the effective green fractions $b_l^{\text{eff}}(k)$, the model can be written as a linear optimization problem, where the node model is replaced using linear inequality constraints [van de Weg et al., 2016].

However, using solely the turn fractions to model the traffic dynamics neglects important non-linear effects. First of all, the impact:

$$\frac{\delta \eta_{l',l}(k)}{\delta u_r^{\text{D}}(k)} \tag{5.41}$$

of the routing decisions $r \in \mathcal{I}_{l'}^{\text{RD}}$ of link l' onto the turn fractions is neglected with $IRD_{l'}$ the set of routing decisions of link l'. Secondly, in a network where traffic is traveling towards certain destinations, changing the control signal of one intersection at time step $k^{\text{p}} < k$ may influence the turn fractions at downstream intersections at later time instants k leading to the following impacts:

$$\frac{\delta \eta_{l',l}(k)}{\delta u_r^{\text{D}}(k^{\text{p}})} \tag{5.42}$$

$$\frac{\delta \eta_{l',l}(k)}{\delta b_y^{\text{eff}}(k^{\text{p}})}\,. \tag{5.43}$$

where $r \in \mathcal{I}_{l''}^{\text{RD}}$ with $\mathcal{I}_{l''}^{\text{RD}}$ the set of routing decision on link l''. Neglecting these effects results in a myopic approximation of the linearization that is less accurate so that it may reduce the performance of the controller. However, including more information requires more computations and hence may increase the required computation time. Therefore, we propose the following procedure to include these effects that may lead to a better trade-off between throughput improvements and computation times. The trade-off will be studied by means of simulations in Section 5.4.

Including these effects into (5.40) gives the following equation for the cumulative link

inflow:

$$N_l^{\text{in}}(k+1) = N_l^{\text{in}}(k) + \sum_{l' \in \mathcal{I}_l^{\text{L,us}}} q_{l'}^{\text{sat}} T^{\text{c}} b_{l'}^{\text{eff}}(k) \Bigg[\eta_{l',l}(k) + \dots \sum_{r \in \mathcal{I}_{l'}^{\text{RD}}} \frac{\delta \eta_{l',l}(k)}{\delta u_r^{\text{D}}(k)} \left(u_r^{\text{D}} - \tilde{u}_r^{\text{D}}(k) \right) + \dots \sum_{l'' \in \mathcal{I}_{l'}^{\text{L,us}}} \sum_{k^{\text{p}} < k} \Bigg[\frac{\delta \eta_{l',l}(k)}{\delta b_y^{\text{eff}}(k^{\text{p}})} \left(b_{l''}^{\text{eff}}(k^{\text{p}}) - \tilde{b}_{l''}^{\text{eff}}(k^{\text{p}}) \right) + \dots \sum_{r \in \mathcal{I}_{yl}^{\text{RD}}} \frac{\delta \eta_{l',l}(k)}{\delta u_r^{\text{D}}(k^{\text{p}})} \left(u_r^{\text{D}}(k^{\text{p}}) - u_r^{\text{D}}(k^{\text{p}}) \right) \Bigg] \Bigg] . \tag{5.44}$$

which is non-linear due to multiplications of the input variables. The variables $\tilde{u}_r^{\text{D}}(k)$ and $\tilde{b}_{l''}^{\text{eff}}(k^{\text{p}})$ are the initial signals of the routing decisions and effective green-fractions as included in the signal $\tilde{U}_i(k)$.

First, this equation is simplified by including the error term $e_{l(k)}$:

$$N_l^{\text{in}}(k+1) = N_l^{\text{in}}(k) + \sum_{l' \in \mathcal{I}_l^{\text{L,us}}} \left(q_{l'}^{\text{sat}} T^{\text{c}} b_{l'}^{\text{eff}}(k) \eta_{l',l}(k) \right) + e_l(k) . \tag{5.45}$$

The vector $\bar{e}_l$ containing all the errors $e_l(k)$ is given as follows:

$$\bar{e}_l = \sum_{l' \in \mathcal{I}_l^{\text{L,us}}} q_{l'}^{\text{sat}} T^{\text{c}} \bar{b}^{\text{eff},0} \Bigg[\sum_{r \in \mathcal{I}_{l'}^{\text{RD}}} J_{l',l',l,r}^{\text{RD}} \left(\bar{\eta}_r^{\text{D}} - \bar{\eta}_r^{\text{D},0} \right) + \dots \sum_{l'' \in \mathcal{I}_{l'}^{\text{L,us}}} \left(J_{l'',jl,l,d}^{\text{b}} \left(\bar{b}_{l''}^{\text{eff}} - \bar{b}_{l''}^{\text{eff},0} \right) + \sum_{r \in \mathcal{I}_{yl}^{\text{RD}}} J_{l'',l',l,r}^{\text{RD}} \left(\bar{\eta}_r^{\text{D}} - \bar{\eta}_r^{\text{D},0} \right) \right) \Bigg] , \tag{5.46}$$

and can be written as a matrix vector multiplication. In (5.46), the Jacobian $J_{l'',l',l,r}^{\text{RD}}$ is a matrix that contains at row i and column j the derivative of the turn-rate $\eta_{l',l}(i)$ with respect to the routing decision $u_r^{\text{D}}(j)$ as given in (5.42). Similarly, the Jacobian $J_{l'',jl,l,d}^{\text{b}}$ is a matrix that describes the derivative of the turn-rate $\eta_{l',l}(i)$ with respect to the effective green fraction $b_{l''}^{\text{eff}}(j)$ as given in (5.43). The method to derive these Jacobian matrices is detailed in 5.A. The vectors $\bar{\eta}_r^{\text{D}}$ and $\bar{b}_{l''}^{\text{eff}}$ contain at every row the routing decision of route r and green fraction of link l'' respectively. The vectors $\bar{\eta}_r^{\text{D},0}$ and $\bar{b}_{l''}^{\text{eff},0}$ are the initial control signals.

The term $e_{l(k)}$ is added to the control input vector $\bar{U}_i$ of the linear optimization problem defined as follows:

$$\bar{U}_i = \begin{bmatrix} \bar{U}_i(k)^\top & \cdots & \bar{U}_i(k+K^{\text{p}}-1)^\top , \end{bmatrix} \tag{5.47}$$

with at every time step the control signal $\bar{U}_i(k)$ defined as:

$$\bar{U}_i(k) = \begin{bmatrix} b_l^{\text{eff}}(k) \\ \vdots \\ b_{n^{\text{L}}}^{\text{eff}}(k) \\ b_o^{\text{eff}}(k) \\ \vdots \\ b_{n^{\text{O}}}^{\text{eff}}(k) \\ u_1^{\text{D}}(k) \\ \vdots \\ u_{n^{\text{decisions}}}^{\text{D}}(k) \\ e_1(k) \\ \vdots \\ e_{n^{\text{L}}}(k) \end{bmatrix} . \tag{5.48}$$

For a given vector $\bar{U}_i$, a linear prediction of the traffic state $\bar{X}$ over the prediction horizon is given as follows:

$$\bar{X} = M^{\text{A}}\bar{X}_0 + M^{\text{B}}\bar{U}_i + M^{\text{C}}\bar{D}\,. \tag{5.49}$$

by multiplying the initial state $\bar{X}_0$, the control signal $\bar{U}_i$, and the disturbance vector $\bar{D}$ with the matrices M^{A}, M^{B}, and M^{C} respectively. The matrices M^{A} and M^{C} are detailed in van de Weg et al. [2016]. The matrix M^{B} is a straightforward extension of the M^{B} matrix detailed in that paper as well. This matrix has to be extended because the matrix used in van de Weg et al. [2016] does not consider the error term $e_{l(k)}$ that is used in the state prediction of this paper. The error term can be computed via linear equality constraints. The vector $\bar{X}$ has the property that for an appropriately chosen row vector $\bar{V}^{\text{TTS}}$, the product $\bar{V}^{\text{TTS}}\bar{X}$ gives the total time spent of all the vehicles in the network.

5.3.4 Linear optimization problem

The next step of the SLP algorithm is finding a search direction vector $\delta\tilde{U}_i$ in which the objective function will decrease. The direction vector $\delta\tilde{U}_i$ is derived from the solution $\bar{U}_i^*$ of the following linear optimization problem:

$$\begin{aligned} \bar{U}_i^* &= arg\ \min_{\bar{U}_i} \bar{V}^{\text{TTS}}\left(M^{\text{A}}\bar{X}_0 + M^{\text{B}}\bar{U}_i + M^{\text{C}}\bar{D}\right), \\ \text{s.t.} \quad \bar{M}^{\text{eq}}\bar{U}_i &= \bar{V}^{\text{eq}}, \\ \bar{M}^{\text{ineq}}\bar{U}_i &\leq \bar{V}^{\text{ineq}} \end{aligned} \tag{5.50}$$

The matrix M^{eq} and vector V^{eq} are used to compute the error terms detailed in (5.46). The matrix $\bar{M}^{\text{ineq}}$ and vector $\bar{V}^{\text{ineq}}$ correspond to the linear inequality constraints that

contain the constraints of (5.34) to (5.36), and constraints to reproduce the dynamics of the LTM. The reader is referred to van de Weg et al. [2016] for a detailed description of these matrices.

Note that the linear model only provides an accurate description of the non-linear model in the vicinity of U_i. Therefore, we limit the search space of the linear optimization problem:

$$U_i - \delta U^{\max} \leq \bar{U}_i \leq U_i + \delta U^{\max}, \tag{5.51}$$

where the vector $\delta U^{\max}$ contains tuning parameters that bound the maximum step-size. The values in the vector $\delta U^{\max}$ have values between 0 and 1, since, they limit the fractions $b_i^{\text{eff}}(k)$ and $u_i^{\text{D}}(k)$, which take values between 0 and 1. A small value for $\delta U^{\max}$ means that the linear model provides a more accurate description of the non-linear model. However, the search space is also limited, so the algorithm might require more iterations, or it might get stuck in a local minimum. On the other hand, by choosing a high value for $\delta U^{\max}$, the solution $\bar{U}_i{}^*$ might be so far from U_i that the linearized model is no longer representative for the non-linear model. This could cause convergence issues of the algorithm. Thus, the parameter vector $\delta U^{\max}$ should be adequately tuned. For the sake of simplicity, all the values in the vector $\delta U^{\max}$ can be chosen the same.

The outcome of the linear optimization problem is the optimal control signal $\bar{U}_i{}^*$. The search direction $\delta\tilde{U}_i$ is given as the difference between $\tilde{U}_i$ and $\tilde{U}_i^*$:

$$\delta\tilde{U}_i = \tilde{U}_i^* - \tilde{U}_i\,. \tag{5.52}$$

The signal $\tilde{U}_i^*$ is derived from the outcome of the linear optimization $\bar{U}_i^*$ by keeping only the effective green fractions of the controlled links and the en-route routing decisions.

5.3.5 Line-search: Computation of the next step

Instead of using the control signal $\bar{U}_i{}^*$ as the control signal U_{i+1} for the next iteration, the control signal is obtained by carrying out a line-search optimization in the direction $\delta\tilde{U}_i$:

$$U_{i+1} = \tilde{U}_i + s^*\delta\tilde{U}_i\,, \tag{5.53}$$

where s^* is the solution to the following optimization problem:

$$\begin{aligned} s^* &= arg\,\min_s Z^{\text{TTS}}(U, D, X_0)\,, \\ \text{s.t. } & 0 \leq s \leq 1\,, \\ & U = \tilde{U}_i + s\delta\tilde{U}_i\,, \\ & M^{\text{ineq}}U \leq V^{\text{ineq}}\,, \end{aligned} \tag{5.54}$$

The constraints used here are identical to the constraints in (5.27). Instead of solving this constrained problem, we eliminate the inequality constraints related to the vector U using penalty functions resulting in the following optimization problem:

$$\begin{aligned} s^* &= arg\ \min_s \bar{Z}^{\mathrm{TTS,ls}}(U, D, X_0)\,, \\ \text{s.t. } 0 &\leq s \leq 1\,, \\ U &= U_i + s(\bar{U_i}^* - U_i)\,, \end{aligned} \tag{5.55}$$

This is a one-dimensional optimization problem that can be solved using a line-search algorithm. In our simulations in Section 5.4 we will use the Fibonacci method as the line-search method. The line-search is stopped when the difference in the objective function from one iteration to another is smaller than ϵ^{ls} (-):

$$|\bar{Z}_i^{\mathrm{TTS,ls}}(X) - \bar{Z}_{i+1}^{\mathrm{TTS,ls}}(X)| \leq \epsilon^{\mathrm{ls}}\,. \tag{5.56}$$

The parameter ϵ^{ls} is a tuning parameter as well. A larger value means that the algorithm might not have reached the optimum. On the other hand, when the value of ϵ^{ls} is decreased, the time to convergence increases. Also, a maximum number of iterations $I^{\mathrm{max,ls}}$ (-) might be included to prevent the line-search algorithm from keeping on iterating when it cannot converge.

5.3.6 Stopping criteria

By solving the above mentioned optimization problem, we find an estimate for the control input U_{i+1} at the next iteration. Before proceeding to the next iteration, it should be checked whether the stopping criterion ϵ^{stop} (-) is satisfied. The stopping criterion is based on the change in the objective function value $Z^{\mathrm{TTS}}(U_i)$ from one iteration to another:

$$|Z^{\mathrm{TTS}}(U_i) - Z^{\mathrm{TTS}}(U_{i+1})| \leq \epsilon^{\mathrm{stop}}\,. \tag{5.57}$$

Since there is no guarantee for convergence, a maximum number of iterations I^{max} (-) is defined, after which the optimization is stopped.

5.3.7 Controller properties and limitations

Due to the analytic approximation of the linearization it is expected that it will take much less time to solve the linear optimization problem when compared to a numerical linearization. Also, it is expected that the inclusion of the error terms leads to a better linearization compared to a linearization solely based on the predicted turn fractions. Nevertheless, the linearization procedure also has several challenges that may need to be addressed in future research.

First of all, it is assumed that the influence of upstream links onto the turn fractions of downstream links propagates with the predicted link travel times. This neglects the fact that there may also be influences that propagate through the network with the queuing dynamics. Hence, the controller may not always be able to find the global optimum. The queuing dynamics may be included by adding matrices that reflect the impact of the future destination link outflow of a link based on the past link outflows. This requires a significant theoretical extension of the framework. Also it may lead to a higher computational complexity of the algorithm.

Secondly, the linearization procedure may become time-consuming because the relations between a lot of paths have to be computed. In the simulation section we will study the impact of the linearization procedure on the computation time and performance by comparing different optimization approaches for two different network sizes.

5.4 Simulation

Simulations are carried out to study the qualitative and quantitative properties of the controller in terms of realized TTS of all the vehicles in the network – including vehicles waiting in the origins – and utilized CPU time. The following steps are carried out to realize this objective:

1. The qualitative behavior of the controller is studied in Section 5.4.2 by analyzing the propagation of traffic in a simple network for a simple demand pattern. It is studied whether the controller is able to distribute the traffic over the network in both free-flow and spill back conditions in such a way that the network throughput is maximized.

2. The quantitative performance of the controller is studied in Section 5.4.3 by comparing the performance of the proposed algorithm with four other optimization approaches for two different networks.

5.4.1 Set-up

Two simple networks, as illustrated in Figure 5.1, are implemented in Matlab R2015a on a computer with a 3.6 GHz processor, 16 Gb RAM, and 8 cores. These grid-shaped network structures are chosen, since they have a similar shape that can easily be extended to a larger network. Every link has the same length of 200 meters – except for link 8 in network 1, which has a length of 800 meters, links 14, 19, and 22 in network 2, which have lengths of 800 meters, and links 2, 7, and 11 in network 2, which have lengths of 400 meters – a free-flow speed of 10 m/s, a shock wave speed of -5 m/s, a jam density of 200 veh/km, and a saturation rate of 2000 veh/h. In network 1,

12 decision variables – 8 green fractions and 4 en-route routing decisions – have to be determined at every time step and in network 2 this mounts up to 42 decision variables – 18 green fractions and 24 en-route routing decisions. The en-route routing decisions are only allowed to vary between 0.1 and 0.9 to ensure that all links are utilized. The simulation period is set to 1800 seconds. The demand patterns used for the different evaluations are shown in Table 5.1 and Table 5.2.

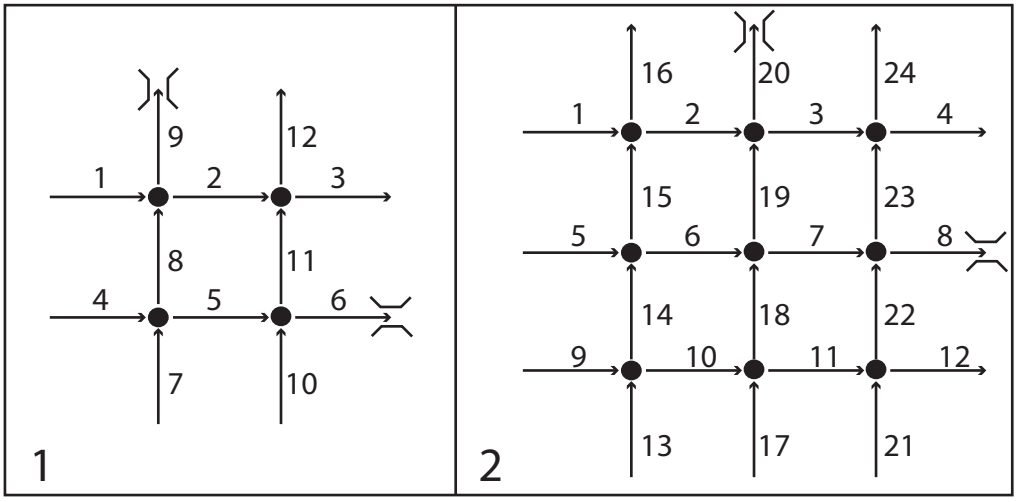


Figure 5.1: The two networks that are used for the evaluations. Note that not all links have the same lengths although the figure suggest otherwise.

The LTM as detailed in Section 5.2 is used as the prediction model in the MPC strategy and the process model that represents the 'real' world. The main difference between the two models is the sampling time which is 10 seconds for the prediction model, and 1 second for the process model. Due to this, the prediction model is more efficient but also less accurate. The controller sampling time step is set to 60 seconds, meaning that every 60 seconds a new optimization is performed and that the control signal is constant during these 60 seconds. The prediction horizon was set to 60 steps.

The algorithm as described in Section 5.3 and referred to as **SLP-I** was implemented in Matlab using the following algorithms and settings. The linear optimization problem detailed in Section 5.3.4 was solved using the 'dual Simplex' algorithm of the 'linprog' function of Matlab. The stopping criteria of the SLP algorithm as detailed in Section 5.3.5 and Section 5.3.6 were chosen as follows: $\epsilon^{\text{ls}} = 1 \cdot 10^{-4}$, $I^{\max,\text{ls}} = 100$, $\epsilon^{\text{stop}} = 5 \cdot 10^{-3}$, and $I^{\max} = 15$, which were found by trial-and-error. The maximum step size of the LP optimization detailed in Section 5.3.4 was set to $\delta U^{\max} = 0.2$.

5.4.2 Qualitative analysis: the behavior of the controller

The qualitative analysis is carried out in order to study the behavior of the algorithm. The main idea is that it is studied to what extent the control action is in accordance with expectations. To this end, network 1 is used together with the demand pattern displayed in Table 5.1. The bottleneck capacity of link 9 is set to 2000 veh/h during the entire simulation and the bottleneck capacity of link 12 is set to 2000 veh/h for the first 900 seconds and after that it is reduced to 50 veh/h. In this way, the traffic will be in free flow during the first 900 seconds. After that, a queue will start to build up in link 6, which can spill-back over time causing delays in links 5 and 10 so that both free-flow and spill-back conditions can be reproduced in one simulation.

Note that link 8 is much longer compared to the other links. Hence, in free-flow conditions the controller will try to send as much traffic towards destinations 3 and 12 over link 5 instead of link 8. However, when the queue in link 6 starts to spill-back towards link 5, it will cause delays in link 5 so that it becomes more efficient to send traffic via link 8. Figure 5.2 A shows the evolution of the en-route routing decisions over time and Figure 5.2 B shows the evolution of the number of vehicles in links 5 and 6 over time. It can be observed that around time 1350 s a queue starts to build up in link 5, in line with expectations, around time 1320 s the controller starts to send traffic from origin 4 via link 8 and around time 1370 s it starts to send traffic from origin 7 via link 8 as well.

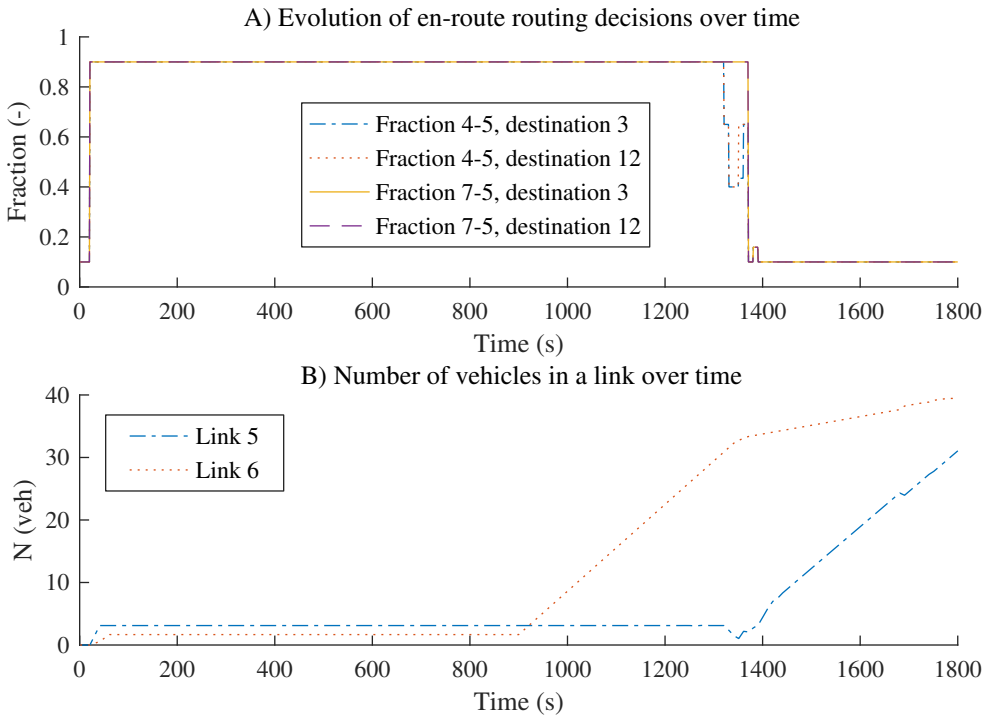


Figure 5.2: A) evolution of the en-route routing decisions over time. B) number of vehicles in links 5 and 6 over time. Around the time when a queue starts to build up in link 5, the controller starts to send traffic to destinations 3 and 12 via link 8.

Table 5.1: O-D demand in veh/h for testing the qualitative behavior

O \ D	9	12	3	6
1	100	100	500	
4	100	100	100	100
7	500	100	100	100
10		500	100	100

5.4.3 Quantitative analysis: comparative analysis

The quantitative analysis is carried out using five different MPC strategies. They all use the same prediction model with a sampling time step of 10 seconds and a prediction horizon of 60 steps. Every controller is given a set of computation time budgets per network type. A simulation is run for every computation time budget. The optimization algorithms are allowed to start from various starting points until the CPU time budget is exceeded. The initial starting point is equal to the control signal obtained at the previous controller time step, if available. In the case that the CPU budget is not exceeded, the optimization is repeated from a new, randomly selected starting point.

The SLP-I algorithm was compared to the following optimization algorithms

SLP-II: The second optimization algorithm is similar to the SLP-I algorithm. However, this algorithm does not consider the impact of past, upstream control signals on the current turn fraction by removing the impact of upstream links in (5.46). In this way it can be studied whether the linearization procedure described in this paper leads to a better trade-off between computation time used and realized throughput.

SLP-I-FP: The third algorithm is similar to the first. However, the line-search step detailed in Section 5.3.5 is skipped by setting $s^* = 1$ in (5.53). In this way, the added value of the line-search step can be studied.

SLP-II-FP: The fourth algorithm is similar to the second. However, the line-search step detailed in Section 5.3.5 is skipped by setting $s^* = 1$. Compared to the SLP-I-FP algorithm, this algorithm does not consider the impact of past, upstream control signals on the current turn fraction.

SQP: The final algorithm tested is the optimization approach called SQP of the 'fmincon' solver that is available in Matlab. This is a commonly used solver for non-linear optimization problems. This algorithm uses a numerical procedure to determine the gradient. A comparison with this algorithm can give insight into the computation time gain when using an analytic linearization. Also, when given sufficient computation time it can give an idea of the maximum achievable performance.

The comparison between the different algorithms enables to study the relative performance. By comparing between the SLP-I and SLP-II algorithm, the added value of taking into account more information when solving the linear optimization problem. It is expected that the SLP-I algorithm may require more time but that it will lead to a better performance. By setting $s^* = 1$ in the SLP-I-FP and SLP-II-FP algorithms, the added value of the line search in the optimization algorithm can be studied. The idea is that the line-search will increase the computation time, but on the other hand may also provide better convergence of the optimization algorithm. The reason for this is that setting $s^* = 1$ may prevent these algorithms from reaching the optimum. Finally, the comparison with the SQP algorithm provides insight into the trade-off between computation time and performance.

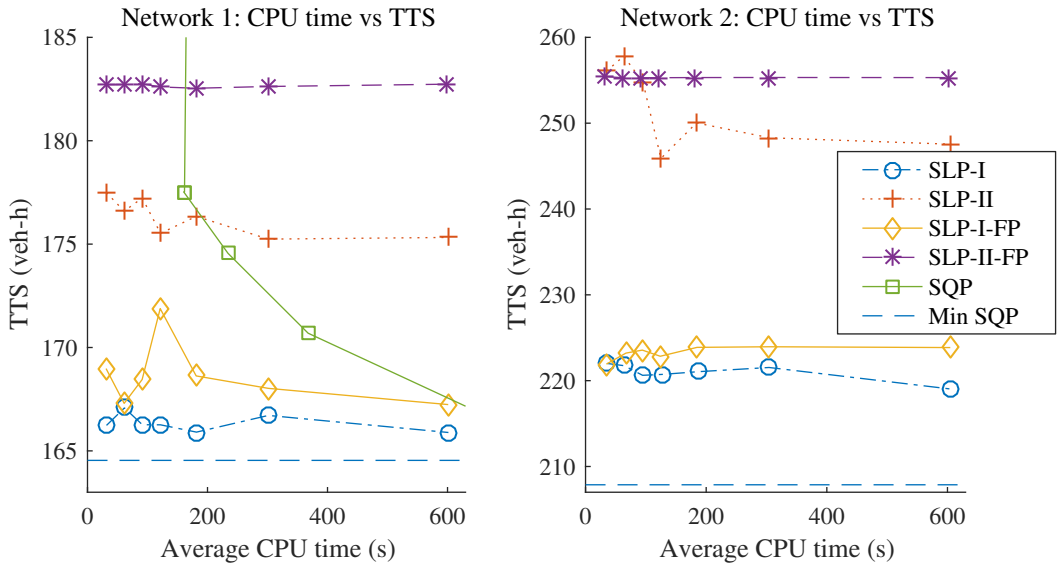


Figure 5.3: Impact of increasing the CPU time budget on the TTS for network 1 (left) and network 2 (right). The blue dashed horizontal line indicates the lowest TTS realized using the SQP algorithm.

The quantitative results are summarized in Figure 5.3 and Table 5.3 for the demand patterns reported in Table 5.2. The table and figure show the realized TTS for different CPU time budgets per iteration. It must be noted that both the CPU time budget and the average CPU time used over the iterations are reported here. The reason for this is that the algorithms cannot be stopped at an exact CPU time budget but only after the budget has been exceeded. The average CPU times are also used in Figure 5.3. It must also be pointed out that for network 2 with the SQP algorithm, only results with a very large CPU time budget are available. The reason for this is that the numerical linearization of the model takes a considerable amount of time so that it was only feasible to run the algorithm from a single starting point per iteration.

Table 5.2: O-D demands in veh/h from time 100 s to 1200 s for testing the quantitative behavior. The left table shows the demand pattern of network 1, and the right table shows the demand pattern of network 2.

Network 1

O \ D	9	12	3	6
1	476	357	360	
4	359	240	243	369
7	354	234	238	363
10		349	352	478

Network 2

O \ D	16	20	24	4	8	12
1	347	313	266	270		
5	304	243	195	200	247	
9	252	191	144	148	195	256
13	251	191	143	147	194	255
17		240	193	197	244	305
21			258	262	309	370

Table 5.3: Comparative results of realized TTS and CPU time used. *This result is not in line with the trend of the other realized TTS values of the SLP-I-FP algorithm. It is caused by the convergence issue of the SLP-I-FP algorithm. **The SQP algorithm as run from 1 starting point without a CPU time budget, hence, this shows the best possible result.

	Network 1									
	SLP-I		SLP-II		SLP-I-FP		SLP-II-FP		SQP	
CPU budget	TTS	CPU	TTS	CPU	TTS	CPU	TTS	CPU	TTS	CPU
30	166.23	31.66	177.50	31.45	168.98	30.94	182.70	30.90	391.54	244.10
60	167.08	61.58	176.59	61.16	167.30	61.12	182.72	60.88	391.54	234.72
90	166.26	91.45	177.23	91.19	168.45	91.07	182.76	90.89	177.53	161.66
120	166.26	121.63	175.52	121.71	171.88*	121.17	182.60	120.62	177.53	160.37
180	165.91	181.64	176.34	181.25	168.63	181.13	182.54	181.06	174.46	234.14
300	166.71	301.51	175.24	301.36	168.03	301.05	182.62	300.98	170.69	367.71
600	165.89	601.50	175.32	601.19	167.24	601.15	182.73	596.85	166.19	702.15
**	-	-	-	-	-	-	-	-	164.54	3276.29
	Network 2									
	SLP-I		SLP-II		SLP-I-FP		SLP-II-FP		SQP	
30	222.00	36.17	256.03	34.35	221.93	34.53	255.42	33.11	-	-
60	221.74	64.12	257.82	63.26	223.21	66.98	255.30	61.57	-	-
90	220.62	94.58	254.74	95.34	223.56	94.15	255.29	92.21	-	-
120	220.73	126.76	245.89	124.99	222.84	123.85	255.30	122.07	-	-
180	221.04	186.34	250.03	183.90	223.88	184.91	255.31	182.10	-	-
300	221.53	304.20	248.30	305.04	223.95	304.62	255.32	302.45	-	-
600	219.03	605.04	247.55	606.37	223.86	604.68	255.31	602.35	-	-
**	-	-	-	-	-	-	-	-	207.87	32585.85

The following observations can be made from the results:

- Both the SLP-I and SLP-I-FP algorithm show a lower TTS for different CPU time budgets compared to the SLP-II and SLP-II-FP algorithms. This indicates the added value of including the impact of the control signal on the turn fractions at downstream links in the future.
- The SLP-I algorithm outperforms the SLP-I-FP algorithm as does the SLP-II algorithm outperform the SLP-II-FP algorithm. This indicates that the inclusion of the line-search step when selecting a new point in the optimization algorithm does lead to better performance. An inspection of the algorithm indeed showed that the FP algorithms do not always converge.
- Extending the CPU time budget of the SLP-I algorithm does not lead to large TTS gains. This indicates that for the selected networks, the controller does not require many starting points to find the optimum, since, the increased CPU time budget mainly results in an increase of the number of starting points explored by the algorithm.
- The TTS of the SLP-I-FP algorithm increases when increasing the CPU time budget. This is probably related to the fact that it does not always converge so that adding more starting point leads to unexpected controller behavior.
- A clear decrease in TTS is visible when extending the CPU time budget of the SQP algorithm. When allowing the SQP algorithm as much time as needed to satisfy the stopping criteria from a single starting point, it is able to realize a better TTS compared to the SLP-I algorithm. This indicates that the SLP-I algorithm does not find the best possible solution. However, it does realize sub-optimal performance in much less CPU time, for instance, the SLP-I algorithm realizes a TTS of 166.23 veh·h in 31.7 seconds while the SQP algorithms requires 702.2 seconds to realize a comparable TTS of 166.19 veh·h for network 1. In the case of network 2, the SQP algorithm requires over 9 hours per iteration to find the optimum of 207.9 veh·h while the SLP-I algorithm is able to achieve a TTS of 220.6 veh·h in 90 seconds which is 6% higher but realized in much less time.

In conclusion, the quantitative results show the added value of the linearization procedure and the line-search step in the SLP optimization algorithm.

5.5 Conclusion and recommendations

This paper proposed an efficient MPC algorithm of the SLP-type for real-time control of en-route decisions and traffic signals. The algorithm is able to realize a better trade-off between computation time and realized throughput when compared to standard

numerical optimization algorithms. This was realized by adopting an efficient traffic flow model, namely the LTM, as the prediction model, and by exploiting an analytic procedure to approximate the linearization of the model in the optimization algorithm.

Evaluations were carried out to assess the quality of the solution found by the algorithm. Qualitative analyses revealed that the control strategy is able to re-route traffic to the shortest paths in free-flow conditions, and that it is able account for the destination-specific flows when distributing the queues over the network in oversaturated traffic regimes. Quantitative analyses that compared the realized TTS for different CPU time budgets showed the added value of the linearization procedure and the use of the line-search step in the optimization algorithm. Although the controller realizes sub-optimal performance, it does realize a better trade-off between CPU time and realized throughput.

Further research can focus on the extension of the algorithm in several ways. This work assumed a 100% compliance to the en-route routing decisions. This assumption may be relaxed by taking the compliance into account in the framework, or the compliance may be realized by providing an (monetary) incentive to the road user. Also, the current algorithm considers all the possible paths between origins and destinations. The algorithm may be extended to only include the relevant paths so that fewer paths have to be studied when linearizing the model. Additionally, a more complex Jacobian may be derived by including not only the actual travel time but also including the propagation of information via the queuing dynamics. It must be noted that the current algorithm already realizes near-optimal performance so that it is not clear whether such an extension will lead to much better performance. Additionally, this is a significant theoretical extension of the framework. Further research can investigate the application in a more practical set-up where, among other things, delays between measurements and actuators are included. Finally, the current algorithm focused on the optimization of aggregated dynamics and a specific network topology. Further research has to be carried out to relax these assumptions.

Acknowledgements

This work is part of the research programme ‘The Application of Operations Research in Urban Transport’, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

5.A Linearization details

The Jacobian $J^{\mathrm{RD}}_{l',l,r}$ of the turn-rate $\eta_{l',l}(k)$ with respect to the routing decision $u^{\mathrm{D}}_r(k)$ that corresponds to destination d is a diagonal matrix with the k^{th} diagonal element

and $k = 1 : K^{\mathrm{P}}$ given as:

$$\frac{\delta \eta_{l',l}(k)}{\delta u_r^{\mathrm{D}}(k)} = \frac{q_{l',d}^{\mathrm{out}}(k)}{q_{l''}^{\mathrm{out}}(k)} \,. \tag{5.58}$$

The Jacobian $J^{\mathrm{RD}}_{l'',l',l,r}$ of the turn-rate $\eta_{l',l}(k)$ with respect to the routing decision $u_r^{\mathrm{D}}(k)$ on link l'' corresponding to destination d is more complex to derive. The reason is that there can be multiple paths $p \in \mathcal{I}^{\mathrm{Paths}}_{l'',l'}$ that lead from link l'' to link l' where the set $\mathcal{I}^{\mathrm{Paths}}_{l'',l'}$ contains all the paths between link l'' and link l'. A Jacobian has to be determined for every path because the travel times on the different path can vary.

Denote with $\mathcal{I}_p^{\mathrm{L,P}}$ the ordered set of links on the path p. The first link in the set $\mathcal{I}_p^{\mathrm{L,P}}$ is link l''. The Jacobian $J^{\mathrm{RD,IN}}_{l'',\mathcal{I}_p^{\mathrm{L,P}}(2),r}$ of the inflow of link $\mathcal{I}_p^{\mathrm{L,P}}(2)$ – i.e., the link directly downstream of link l'' – oriented to destination d with respect to the routing decision $u_r^{\mathrm{D}}(k)$ is a diagonal matrix of which the k^{th} diagonal element defined as:

$$\frac{\delta q^{\mathrm{in}}_{\mathcal{I}_p^{\mathrm{L,P}}(2),d}}{\delta u_r^{\mathrm{D}}} = q^{\mathrm{out}}_{l'',d} \,. \tag{5.59}$$

Next, define the following relation between the inflow of link $\mathcal{I}_p^{\mathrm{L,P}}(2)$ and the outflow of link $l' = \mathcal{I}_p^{\mathrm{L,P}}(n_p^{\mathrm{L,P}})$ with $n_p^{\mathrm{L,P}}$ the number of links on path p:

$$\bar{q}_{l'}^{\mathrm{out}} = M_{l'}^{\mathrm{tr}} \prod_{l''=2}^{n_p^{\mathrm{L,P}}-1} \left(M^{\mathrm{D}}_{\mathcal{I}_p^{\mathrm{L,P}}(l''),\mathcal{I}_p^{\mathrm{L,P}}(l''+1),d} M^{\mathrm{tr}}_{\mathcal{I}_p^{\mathrm{L,P}}(l'')} \right) \bar{q}_{l'}^{\mathrm{in}} \,. \tag{5.60}$$

Here, the matrix M_l^{tr} maps the inflow vector $\bar{q}_l^{\mathrm{in}}$ (veh/h) of link l to the outflow vector $\bar{q}_l^{\mathrm{out}}$ (veh/h)of link l so that:

$$\bar{q}_l^{\mathrm{out}} = M_l^{\mathrm{tr}} \bar{q}_l^{\mathrm{in}} \,, \tag{5.61}$$

and the matrix $M^{\mathrm{D}}_{l',l,d}$ maps the outflow vector $\bar{q}^{\mathrm{out}}_{l',d}$ oriented to destination d on link l to the inflow vector $\bar{q}^{\mathrm{in}}_{l,d}$ oriented to destination d on link l:

$$\bar{q}^{\mathrm{in}}_{l,d} = M^{\mathrm{D}}_{l',l,d} \bar{q}^{\mathrm{out}}_{l',d} \,. \tag{5.62}$$

Now, we can find the Jacobian $J^{\mathrm{RD,OUT}}_{yl,l',r}$ of which the element on row i and column j is defined as:

$$\frac{\delta q_{l'}^{\mathrm{out}}(i)}{\delta u_r^{\mathrm{D}}(j)} \,, \tag{5.63}$$

i.e., the derivative of the outflow from link l' with respect to the derivative of the en-route decision d on link l''. This Jacobian is found by adding the derivatives between the two links over the different paths:

$$J^{\mathrm{RD,OUT}}_{yl,l',r} = \sum_{p \in \mathcal{I}^{\mathrm{Paths}}_{l'',l'}} \left(M_{l'}^{\mathrm{tr}} \prod_{l''=2}^{n_p^{\mathrm{L,P}}} \left(M^{\mathrm{D}}_{\mathcal{I}_p^{\mathrm{L,P}}(l''),\mathcal{I}_p^{\mathrm{L,P}}(l''+1),d} M^{\mathrm{tr}}_{\mathcal{I}_p^{\mathrm{L,P}}(l'')} \right) J^{\mathrm{RD,IN}}_{yl,\mathcal{I}^{\mathrm{L,P}}(2),r} \right) . \tag{5.64}$$

Using this Jacobian, it becomes possible to compute the Jacobian $J^{\mathrm{RD}}_{l'',l',l,r}$, which is given as:

$$J^{\mathrm{RD}}_{l'',l',l,r} = J^{\mathrm{turn,OUT}}_{l',l,d} J^{\mathrm{RD,OUT}}_{l'',l',l,r} \,, \tag{5.65}$$

with the Jacobian $J^{\mathrm{turn,OUT}}_{l',l,d}$ a matrix where the k^{th} diagonal element is given as:

$$\frac{\delta \eta_{l',l}(k)}{\delta q^{\mathrm{out}}_{l',d}(k)} \,. \tag{5.66}$$

Similarly, the Jacobian $J^{\mathrm{b}}_{l'',l',l}$ of the derivative of turn-fraction form link l' to link l to the effective green fraction on link l'' is given as:

$$J^{\mathrm{b}}_{l'',l',l} = \sum_{d \in \mathcal{I}^{\mathrm{D}}_{l''}} J^{\mathrm{turn,OUT}}_{l',l,d} \ldots \tag{5.67}$$

$$\sum_{p \in \mathcal{I}^{\mathrm{Paths}}_{l'',l'}} \left(M^{\mathrm{tr}}_{l'} \prod_{l''=2}^{n^{\mathrm{L,P}}_p} \left(M^{\mathrm{D}}_{\mathcal{I}^{\mathrm{L,P}}_p(l''),\mathcal{I}^{\mathrm{L,P}}_p(l''+1),d} M^{\mathrm{tr}}_{\mathcal{I}^{\mathrm{L,P}}_p(l'')} \right) M^{\mathrm{D}}_{l'',\mathcal{I}^{\mathrm{L,P}}(2),d} J^{\mathrm{b,OUT}}_{yl,d} \right) .$$

The Jacobian $J^{\mathrm{b,OUT}}_{yl,d}$ is a diagonal matrix of which the k^{th} diagonal element is given as:

$$\frac{\delta q^{\mathrm{out}}_{l'',d}(k)}{\delta b^{\mathrm{eff}}_{l''}(k)} = \frac{q^{\mathrm{out}}_{l'',d}(k)}{q^{\mathrm{out}}_{l''}(k)} q^{\mathrm{sat}}_{l''} \,. \tag{5.68}$$

5.B Overview of variables

In order to provide the reader with a quick overview of the different variables used in the following sections we provide an overview of the mathematical notation:

- Timing
  - The model that is used in this paper is a discrete-time model. To this end, the time step k (-) refers to the period $\big[Tk, T(k+1)\big)$ (h) where T (h) is the model sampling time.
  - The index k^{free} (-) represents the number of time steps needed to travel through the link in free-flow conditions. The fraction γ^{free} (-) is the fraction of a time step that k^{free} exceeds the free-flow travel time t^{free} (h) so that $t^{\mathrm{free}} = (k^{\mathrm{free}} + \gamma^{\mathrm{free}})T$.
  - The index k^{shock} (-) represents the number of time steps needed for a shock wave to travel through the link. The fraction γ^{shock} (-) is the fraction of a time step that k^{shock} exceeds the shock wave travel time t^{shock} (h) so that $t^{\mathrm{shock}} = (k^{\mathrm{shock}} + \gamma^{\mathrm{shock}})T$.

  - The index $k^{\mathrm{tr}}(k)$ (-) represents the number of time steps that a vehicle that exits the link at time step k needed to travel through the link. The fraction $\gamma^{\mathrm{tr}}(k)$ (-) is the fraction of a time step that $k^{\mathrm{tr}}(k)$ exceeds the travel time $t^{\mathrm{tr}}(k)$ (h) so that $t^{\mathrm{tr}}(k) = (k^{\mathrm{tr}}(k) = \gamma^{\mathrm{tr}}(k))T$.
  - The number of time steps of the prediction horizon is denoted by K^{p} (-)

- The sets and indexes used are listed below. Note that sets are referred to with the symbol $\mathcal{I}$:
  - Links are referred to as $l \in \mathcal{I}^{\mathrm{L}}$ (-) where $\mathcal{I}^{\mathrm{L}}$ is the set of all links in the network
  - Origins are referred to as $o \in \mathcal{I}^{\mathrm{O}}$ (-) where $\mathcal{I}^{\mathrm{O}}$ is the set of all origins in the network
  - Destinations are referred to as $d \in \mathcal{I}^{\mathrm{D}}$ (-) where $\mathcal{I}^{\mathrm{D}}$ is the set of all destinations in the network.
  - Routing decisions are referred to as $r \in \mathcal{I}^{\mathrm{RD}}$ (-) where $\mathcal{I}^{\mathrm{RD}}$ is the set of all the routing decisions
  - Nodes are referred to as $m \in \mathcal{I}^{\mathrm{N}}$ (-) where $\mathcal{I}^{\mathrm{N}}$ is the set of all nodes in the network
  - Conflicts between links on a node are referred to with index c (-)
  - Paths between links l and l' are referred to as $p \in \mathcal{I}_{l,l'}^{\mathrm{Paths}}$ (-) where $\mathcal{I}_{l,l'}^{\mathrm{Paths}}$ is the set of all paths between links l and l'
  - The set $\mathcal{I}_{p}^{\mathrm{L,P}}$ contains all the link indexes on path p

- In the case that a distinction has to be made between indexes of the same type, an accent is used, for instance, for instance, link l and link l'.

- Variables
  - The variables $N^{\mathrm{in}}(k)$ (veh) and $N^{\mathrm{out}}(k)$ (veh) denote the cumulative inflow and outflow of origins and links
  - The number $N_l^{\max}$ (veh) is the maximum number of vehicles that fits in a link
  - The factor $\beta_l(k)$ (-) is the reduction factor of the demand to account for an outflow reduction due to spill-back
  - The fraction $b_l(k)$ (-) is the fraction of green time given to a link l
  - The fraction $b_l^{\mathrm{eff}}(k)$ is the fraction of the time that green is effectively used by the link
  - The flow q_l^{sat} (veh/h) is the saturation rate, i.e. the link outflow capacity
  - The flow q_l^{in} (veh/h) is the inflow of link l

  - The flow $q_l^{\text{in,sp}}(k)$ (veh/h) is the maximum receiving flow of link l due to spillback
  - The flow q_l^{out} (veh/h) is the outflow of link l
  - The flow $q_l^{\text{out,max}}(k)$ (veh/h) is the maximum allowed outflow of a link
  - The flow q_o^{in} (veh/h) is the demand at origin o. The flow $q_{o,d}^{\text{in}}$ is the demand bound to destination d at origin o.
  - The flow $q_{l,d}^{\text{out}}(k)$ (veh) indicates the outflow of link l bound to destination d.
  - The variable $\bar{u}_{l,l',d}^{\text{D}}(k)$ (-) indicates the fraction of the flow $q_{l,d}^{\text{out}}(k)$ (veh) moving to downstream link l'.
  - The variable $u_r^{\text{D}}(k)$ (-) is the en-route decision variable.
  - The symbol e is used for errors between the states predicted by the non-linear model and the linear model.
  - The variable Z^{TTS} (veh·h) expresses the total time spent (TTS) by all the vehicles in the network
  - The vector X contains a prediction of the traffic state
  - The vector X_0 contains the initial traffic state
  - The vector D contains the prediced disturbances
  - The vector U contains a candidate control signal
  - The vector $\tilde{U}$ contains the effective value of a candidate control signal
  - The symbol M is used to indicate matrices
  - The symbol V is used to indicate vectors
  - The symbol J is used to indicate a Jacobian matrix
  - The fraction s (-) is the step-size taken in the line-search step of the algorithm
  - The thresholds ϵ^{stop} (-) and ϵ^{ls} (-) are used as stopping criteria for the optimization algorithms
  - The parameters $I^{\max}$ (-) and $I^{\max,\text{ls}}$ (-) are used as maximum numbers of iterations of the optimization algorithms

- In some cases a bar $\bar{\cdot}$ is placed over a variable to indicate that the variable is used in the linear optimization problem.

Chapter 6

Hierarchical Control Framework for Coordination of Intersection Signal Timings in all Traffic Regimes

This chapter proposes a hierarchical control framework for the improvement of urban network throughput. The hierarchical control framework realizes a translation of the optimized control signal of the MPC strategy proposed in Chapter 4 to actual signal timings. This chapter is based on the following paper that is currently under review:

G.S. van de Weg, H.L. Vu, A. Hegyi, and S.P. Hoogendoorn, A Hierarchical Control Framework for Coordination of Intersection Signal Timings in all Traffic Regimes. *Transactions on Intelligent Transportation Systems*, submitted 2017-4-13.

Abstract

In this paper we develop a hierarchical approach to optimize the signal timings in an urban traffic network taking into account the different dynamics in all traffic regimes. The hierarchical control framework consists of two layers. The network coordination layer uses a model predictive control strategy based on a simplified traffic flow model to provide reference outflow trajectories. The reference outflow trajectories represent average link outflows over a time horizon which could be simultaneously nonzero for conflicting directions and which require to be mapped to a green-red switching signal that can be applied to traffic lights. To this end, the individual intersection control layer then selects at every individual intersection the signal timing stage that realizes an outflow which has the smallest error with respect to the reference outflow trajectory. The proposed framework is tested using both macroscopic and microscopic simulation. It is shown that the control framework can outperform a greedy control policy

that maximizes the individual intersection outflows, and that the control framework can distribute the queues over the network in a way that the network outflow is improved. Simulations using a macroscopic model allow the direct application of the reference outflows computed by the network coordination layer, the results indicate that the mapping of the reference outflows to the detailed signal timings by the individual intersection control layer only introduces a small performance loss.

6.1 Introduction

Coordination of the signal timings of intersections to improve the performance of urban traffic networks is a complex problem. One of the main reasons for this is that coordination requires accounting for the impact of the signal timings on the propagation of traffic over the network. This introduces several issues as discussed below.

One of the main issues of controlling signal timings plans is that they have a switching structure, meaning that a stage – i.e., a set of streams that can be active simultaneously – can either be green or red. This introduces interruptions (or discontinuities) in the traffic flows at intersections. Due to these discontinuities, optimizing the signal timing plans results in a mixed integer optimization problem that is difficult to solve. This is problematic, since only a limited amount of computation time is available for the real-time application of traffic control strategies. Additionally, other properties of the signal timing plan such as clearance times, offsets, (predetermined) stage sequences, and cycle times, add to the complexity.

Apart from that, the direction of the interaction between intersections changes when the traffic regime changes as discussed in [van de Weg et al., 2016]. More specifically, in the undersaturated regime – i.e., when queues are completely emptied during a green time period – an increase in the outflow of an upstream intersection can lead to a change in the outflow at a downstream intersection. This relation is typically used in green-wave approaches that allow vehicles to pass multiple intersections without stopping. In the saturated regime – i.e., when queues neither become empty, nor will spill back to upstream intersections – there is no such strong coupling. Finally, in the oversaturated regime – i.e., when queues spill back to upstream intersections – a change in the outflow at a downstream intersection leads to a change in the outflow of an upstream intersection at a later time instant. All these effects have to be taken into account when optimizing the timing of a signal controller.

The aim of this paper is to design a control strategy for the coordination of signal timings of multiple intersections. The control strategy has to account for all the traffic regimes. It also has to be real-time feasible, meaning that it can compute the control actions within the controller sampling time. The controller sampling time is the time period between updates of the control signal, which is typically in the range of one to several minutes.

6.1.1 Literature

This section discusses approaches to the urban traffic network control problem. We examine for what traffic regimes the different strategies are designed, whether they are real-time feasible, and in what way signal timings are considered. First, various well-known or recent control strategies are discussed. After that, the review focuses on model-based predictive control strategies.

Approaches to the urban traffic network control problem

The first approaches to the coordination of intersections focused on performance improvement in the undersaturated traffic regime. A well-known example is the MAXBAND approach proposed by Little [1966] for the creation of green-waves between intersections. MAXBAND computes the signal timings off-line in such a way that traffic can pass multiple intersection without stopping. A disadvantage of off-line control is that it cannot adapt to changes in the traffic demand. SCOOT [Hunt et al., 1982] and SCATS [Luk et al., 1982] are examples of widely used control strategies for undersaturated traffic regimes that can dynamically adjust to changes in the traffic situation. The performance of SCOOT may deteriorate in saturated and oversaturated regimes according to Papageorgiou et al. [2003]. Recently, Lämmer and Helbing [2008] proposed a decentralized algorithm that decides at each time instant which stage to actuate in order to reduce the delay at every intersection in the undersaturated regime.

Diakaki et al. [2003] proposed the TUC algorithm, which is specifically designed to improve the urban traffic network throughput in the saturated regime. TUC has a feedback structure, and adjusts the green times at an intersection based on the queue lengths in the network. Various extensions to TUC have been proposed, such as the inclusion of green-waves [Kraus Jr et al., 2010]. Recently, the max-pressure (or back-pressure) algorithm was proposed to address the coordination problem in the saturated regime [Varaiya, 2013, Le et al., 2015]. The max-pressure algorithm decides at every time instant which stage to actuate. This decision is made using information on the queues located directly upstream and downstream of the intersection, so that no centralized communication structure is required.

The performance of the aforementioned control strategies may deteriorate in the oversaturated regime, since the impact of spill back and the corresponding shock wave dynamics are not considered in the controller design. In that regime, congestion may propagate through the network causing a loss of efficiency at intersections and potentially leading to gridlock [Daganzo, 2007]. One way to address this issue is by perimeter control based on the network fundamental diagram (NFD) [Keyvan-Ekbatani et al., 2012]. The aim of this strategy is to keep the number of vehicles in the network below or at the critical density of the network fundamental diagram so that congestion is prevented. An issue with this approach is that the shape of the NFD may be affected by

the intersection control strategies.

In conclusion, all these approaches are designed to improve the performance in only one or two of the three traffic regimes. A promising approach to include all the traffic regimes is the application of a predictive control strategy. However, this is a challenging task, as discussed in the next section.

Model-based predictive control approaches

Model predictive control (MPC) is a popular method to determine a control action that accounts for the long-term impact of a control signal on the system's performance. It is typically used to determine a control signal over a period of time called the control horizon, that optimizes the performance over a period of time called the prediction horizon [Garcia et al., 1989, Mayne et al., 2000]. MPC is a procedure in which the impact – expressed using an objective function – of a candidate control signal on the propagation of traffic over the network is predicted using a prediction model. At every controller sampling time instant, the control signal that optimizes the objective function is recomputed using the most recent traffic state measurements. This is commonly referred to as the receding horizon principle.

Lo [1999] and Van den Berg et al. [2007] have proposed MPC approaches for the optimization of signal timings. Lo [1999] used the Cell-Transmission Model (CTM) to predict the traffic dynamics, and modelled the signal timings using binary variables – i.e., a stream can receive either green (1) or red (0). This resulted in a mixed-integer linear programming problem (MILP). Van den Berg et al. [2007] used the horizontal queuing model of Kashani and Saridis [1983] to model all the traffic regimes, resulting in a non-linear optimization problem. Lin et al. [2011] used the S-model, which is a simplification of the model of Van den Berg et al. [2007], to formulate another MILP optimization problem. Despite the ability to explicitly consider signal timings and all traffic regimes, all of the resulting non-linear and MILP optimization problems are cumbersome to solve. Due to this, these methods are not real-time feasible when applied to medium to large-scale networks of several (tens of) intersections.

The scalability problem can be mitigated by aggregating the traffic dynamics to (several) tens of seconds and replacing the binary signal timings with average outflows so that continuous or linear optimization problems can be formulated [van de Weg et al., 2016, Aboudolas et al., 2010, Le et al., 2013]. Aboudolas et al. [2010] proposed a linear MPC approach based on the store-and-forward model for the saturated regime which resulted in a drastic reduction of the computation time. Le et al. [2013] proposed an MPC approach based on a modified version of the CTM for undersaturated and saturated regimes. Recently, van de Weg et al. [2016] proposed the use of the Link Transmission Model (LTM) in a linear MPC framework. This approach is capable of reproducing all traffic regimes and is real-time feasible. However, none of these methods consider signal timings, so they are not directly applicable to a real traffic network.

6.1.2 Research approach and contributions

This paper develops a real-time feasible, hierarchical control framework for the control of signal timings in order to improve the urban network throughput in all traffic regimes. The main contribution of the research is the design of a real-time feasible framework for the control of signal timings that can optimize the distribution of traffic over a network while taking into account the upstream propagating waves caused by spillback.

The hierarchical control framework consists of two layers. The top layer – called the network coordination layer – consists of the linear MPC strategy for urban traffic networks (LML-U) of van de Weg et al. [2016] that optimizes the aggregated traffic dynamics. The LML-U strategy distributes the traffic over the network so that the average throughput is maximized over a time horizon. In this paper, the optimized control signal is translated to near-future reference outflows for the entire time horizon of the links in the network. The reference outflow trajectories cannot be directly applied to the network since they represent average traffic flows while traffic lights require a green-red switching signal. The bottom layer – called the individual intersection layer – consists of the local intersection controllers. The goal of these controllers is to select the stage at every time step that minimizes the error with the reference outflows. The framework is designed in such a way that control strategies other than the one implemented in this paper may be used in both the top and bottom layers. The proposed framework is evaluated using simulation experiments.

The second contribution of the paper is to show that compared to locally optimizing the intersection outflows, the resulting control strategy can improve the throughput by distributing traffic over the network in spillback conditions. This is shown quantitatively by comparing the proposed strategy to a strategy that optimizes the local intersection outflows, and qualitatively by studying the realized traffic states.

The third contribution of the paper is to provide insight into the controller performance when varying the controller sampling times and when applied to different process models. The reason why this is studied is that an important issue of MPC strategies is that the mismatch between the prediction and process model may negatively affect the controller performance. One way to limit the impact of this mismatch is by reducing the sampling time of the controller, so that the possible prediction errors can be corrected more frequently by using new measurements. In the proposed framework, the sampling times of the two layers can be varied, both of which may affect the controller performance. Reducing the sampling time of the individual intersection layer allows more frequent switching, leading to a better tracking of the reference outflow trajectories; reducing the sampling time of the network coordination layer allows for a more frequent correction of prediction errors. Qualitative analyses are carried out in which the sampling times of the different layers are varied. In addition, simulations are carried out with two different process models, namely, the LTM and the microscopic model Vissim that has a larger mismatch with the prediction model.

6.1.3 Design considerations

Several factors were considered when designing the control strategy in order to simplify the problem or to emphasize the most important control features.

As stated before, an intersection control program is rather complex. To simplify this, we assume that there is no fixed stage sequence. Also, no minimum green times, and no fixed cycle times are used. Clearance times – i.e., the time used to clear the intersection between two conflicting stages – are included in the approach.

The control strategy has to be real-time feasible. This means that the time it takes to compute the control signal is shorter than the controller sampling time, which is typically in the range of one to several minutes. A longer controller sampling time is beneficial, since it allows more time to optimize the control signal. However, the controller sampling time should be kept short so that the controller can quickly respond to traffic changes and unexpected events.

The aim of the controller is to improve the throughput. In practice, other performance indicators might also be included, such as equity, pollution, and reliability. Their inclusion, however, is beyond the scope of this paper.

Finally, the paper focuses on networks used solely by motorized traffic. The extension to networks used by heterogeneous traffic – e.g. cars, trucks, public transport, and bicycles – is left for further research.

6.2 Controller design

In order to bridge the gap between the high computation time required by optimization based control strategies and the low computation time, but lower expected performance, of feedback-based control strategies, a hierarchical control framework is proposed in this paper. The framework is presented in Figure 6.1 and consists of two layers:

1. The top layer uses an aggregated prediction model to optimize the network throughput every T^{ref} seconds, where T^{ref} is in the range of one to several minutes. The control signal consists of the fractions of green time that every stream in the network has to realize, but which are not directly applicable by the traffic signal controllers. Nevertheless, the desired behavior of the traffic system – for instance, a prediction of link outflows – can be derived from this signal. Hence, reference outflow trajectories can be derived from the optimized signal, such as the reference cumulative outflow of a link, or a reference number of vehicles that has to be present in the link.

2. The bottom layer consists of the local intersection controllers. The task of the local intersection controllers is to track the reference outflows. This is realized

by selecting every T^{local} seconds – in the range of 5 to 10 seconds – the stage that is expected to lead to the smallest reference tracking error in the next T^{local} seconds. The local intersection controllers may not be able to track the reference outflows exactly, because they were determined using a simplified traffic flow model. However, it is expected that the average behavior of the local intersection controllers will lead to improved network performance when the tracking error remains small.

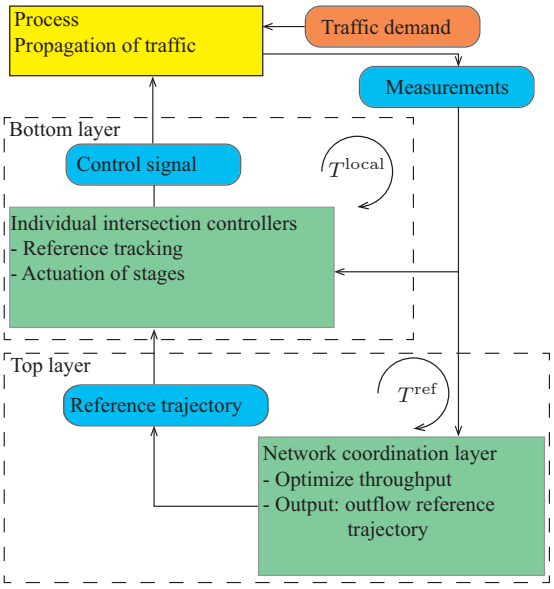


Figure 6.1: Schematic overview of the control strategy

The advantage of this framework is that the signal timings are determined in a decentralized way; i.e., every intersection requires only measurements of the direct upstream and downstream links. However, due to the tracking of the reference outflows, the individual intersection controllers are capable of realizing network-wide performance improvements.

The idea behind the proposed framework is that different control algorithms can be applied to the different layers. In this way, the framework can be adapted to different traffic networks, situations, and desired controller properties. As a proof-of-concept, Section 6.2.2 details the implementation of a linear MPC strategy – called LML-U – based on the link transmission model in the coordination layer, and Section 6.2.3 presents a greedy reference tracking (GRT) strategy for the individual intersection controller layer. Hence, the proposed strategy is called LML-U + GRT. In Section 6.3, simulation results of this implementation are presented.

6.2.1 Timing

Discrete timing is considered in this paper. The time step k (-) and sampling time T (s) refer to the period $t \in \big[Tk, T(k+1)\big)$ (s). It is assumed that the sampling time of

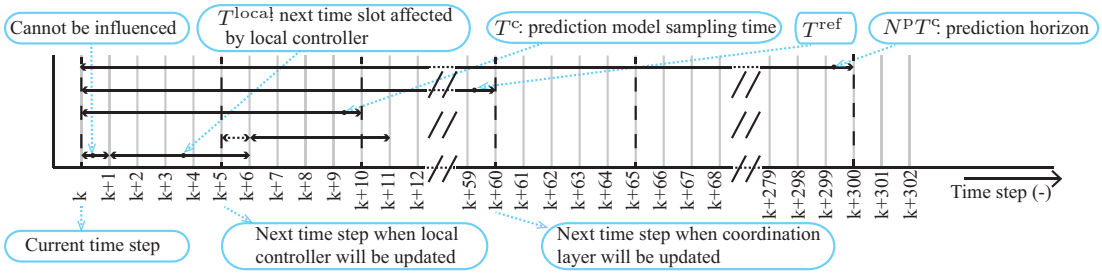


Figure 6.2: Schematic overview of the timing used. In this example, the sampling time T is 1 second, the intersection controller sampling time T^{local} is 5 seconds, the prediction model sampling time T^{c} is 10 seconds, the coordination layer sampling time T^{ref} is 60 seconds, and the prediction horizon N^{p} is 30 steps.

the measurements is equal to T. The prediction model has a sampling time step k^{c} (-) and sampling time T^{c} (s). It holds that $T^{\text{c}} = \epsilon^{\text{c}} T$ with the factor $\epsilon^{\text{c}} \in \mathcal{Z}^+$ – i.e., it is a strictly positive integer. The intersection controllers select a new stage to actuate every controller sampling time step k^{local} (-) with controller sampling time step T^{local} (s) for which it holds that $T^{\text{local}} = \epsilon^{\text{local}} T$, with the factor $\epsilon^{\text{local}} \in \mathcal{Z}^+$. The reference outflow trajectory is updated every time step k^{ref} (-) with the sampling time step $T^{\text{ref}} = \epsilon^{\text{ref}} T$ seconds, with $\epsilon^{\text{ref}} \in \mathcal{Z}^+$. It also holds that $T^{\text{ref}} = \epsilon^{\text{c,ref}} T^{\text{c}}$, with $\epsilon^{\text{c,ref}} \in \mathcal{Z}^+$. It follows that $k = (k^{\text{local}} - 1)\epsilon^{\text{local}} + 1 = (k^{\text{c}} - 1)\epsilon^{\text{c}} + 1 = (k^{\text{ref}} - 1)\epsilon^{\text{ref}} + 1$, and that $k^{\text{c}} = (k^{\text{ref}} - 1)\epsilon^{\text{c,ref}} + 1$. Figure 6.2 provides an overview of the timing used in this paper.

It must be noted that a measurement that is available at time step k reflects the traffic state at the beginning of the time period k. It is thus not possible to change the control action at time step k. Hence, at time step k the control signal for the next time step $k + 1$ will be determined. So, in this paper the control action at time step k^{local} is determined based on the data available at time step $(k^{\text{local}} - 1)\epsilon^{\text{local}} = k$.

6.2.2 Network coordination layer: LML-U approach

The task of the network coordination layer – i.e., the top layer of the proposed framework – is to determine the reference outflows that optimize the network throughput. Recall that the coordination layer sampling time T^{ref} (s) is in the range of one to several minutes. Hence, in order to satisfy real-time feasibility, the coordination layer has to be able to compute the reference outflow trajectories within one to several minutes.

To this end, the recently developed linear model predictive control strategy using the link transmission model for urban traffic networks (LML-U) is chosen in the coordination layer [van de Weg et al., 2016]. This approach has the advantage that it considers all relevant first-order traffic dynamics – i.e., upstream and downstream propagating waves – using only two traffic states. Compared to segment-based models, such as the CTM, this is more efficient from a computational point of view. The approach requires a prediction of the traffic demand, turn-fractions, and maximum network outflows. Its

output consists of the optimized fractions of green time used by the traffic streams in the network. The remainder of this section first discusses the prediction model used in more detail, next the optimization problem is introduced, and finally the approach to compute the reference outflow trajectories from the optimization output is presented.

The prediction model

The prediction model used in the LML-U control strategy is the LTM. The main elements used here are links – indicated with index i^{L} (-) – and origins – indicated with index i^{O} (-). The traffic dynamics of origins and links are updated using two traffic states; the cumulative link inflow $N^{\mathrm{in}}_{i^{\mathrm{L}}}(k^{\mathrm{c}})$ (veh) and outflow $N^{\mathrm{out}}_{i^{\mathrm{L}}}(k^{\mathrm{c}})$ (veh), and the cumulative origin inflow $N^{\mathrm{O,in}}_{i^{\mathrm{O}}}(k^{\mathrm{c}})$ (veh) origin outflow $N^{\mathrm{O,in}}_{i^{\mathrm{O}}}(k^{\mathrm{c}})$ (veh). Every outflow is controlled using a control parameter $b^{\mathrm{eff}}_{i^{\mathrm{L}}}(k^{\mathrm{c}})$ for links and $b^{\mathrm{eff,O}}_{i^{\mathrm{O}}}(k^{\mathrm{c}})$ for origins that expresses the effective fraction of green time used during the time step k^{c}. Note that this optimization approach is presented in more detail in van de Weg et al. [2016]. The interested reader is referred to [Yperman, 2007] for a more detailed description of the LTM.

The cumulative link outflow is updated using the following equation:

$$N^{\mathrm{out}}_{i^{\mathrm{L}}}(k^{\mathrm{c}}+1) = N^{\mathrm{out}}_{i^{\mathrm{L}}}(k^{\mathrm{c}}) + q^{\mathrm{sat}}_{i^{\mathrm{L}}} T^{\mathrm{c}} b^{\mathrm{eff}}_{i^{\mathrm{L}}}(k^{\mathrm{c}})\,, \tag{6.1}$$

where $q^{\mathrm{sat}}_{i^{\mathrm{L}}}$ (veh/h) is the saturation rate. The cumulative link inflow is modeled as the sum of the outflows of upstream links $j^{\mathrm{L}} \in \mathcal{I}^{\mathrm{L,us}}_{i^{\mathrm{L}}}$ and origins $i^{\mathrm{O}} \in \mathcal{I}^{\mathrm{O,us}}_{i^{\mathrm{L}}}$ multiplied with the turn-fractions $\eta_{j^{\mathrm{L}},i^{\mathrm{L}}}(k)$ given as:

$$\begin{aligned} N^{\mathrm{in}}_{i^{\mathrm{L}}}(k^{\mathrm{c}}+1) = N^{\mathrm{in}}_{i^{\mathrm{L}}}(k^{\mathrm{c}}) + & \sum_{j^{\mathrm{L}} \in \mathcal{I}^{\mathrm{L,us}}_{i^{\mathrm{L}}}} \left(\eta_{j^{\mathrm{L}},i^{\mathrm{L}}}(k^{\mathrm{c}}) b^{\mathrm{eff}}_{i^{\mathrm{L}}}(k^{\mathrm{c}}) q^{\mathrm{sat}}_{i^{\mathrm{L}}} T^{\mathrm{c}} \right) + \dots \\ & \sum_{i^{\mathrm{O}} \in \mathcal{I}^{\mathrm{O,us}}_{i^{\mathrm{L}}}} \left(\eta_{i^{\mathrm{O}},i^{\mathrm{L}}}(k^{\mathrm{c}}) b^{\mathrm{eff,O}}_{i^{\mathrm{O}}}(k^{\mathrm{c}}) q^{\mathrm{cap}}_{i^{\mathrm{O}}} T^{\mathrm{c}} \right), \end{aligned} \tag{6.2}$$

where the set $\mathcal{I}^{\mathrm{L,us}}_{i^{\mathrm{L}}}$ is the set of links directly upstream of link i^{L} and the set $\mathcal{I}^{\mathrm{O,us}}_{i^{\mathrm{L}}}$ is the set of origins directly upstream of link i^{L}. The fraction $\eta_{j^{\mathrm{L}},i^{\mathrm{L}}}(k^{\mathrm{c}})$ indicates the turn fraction form link j^{L} to link i^{L}, and the fraction $\eta_{i^{\mathrm{O}},i^{\mathrm{L}}}(k^{\mathrm{c}})$ (-) indicates the turn fraction form origin i^{O} to link i^{L}.

In order to model free-flow dynamics, the cumulative link outflow is bound from above, so that vehicles cannot travel through the link faster than the free flow travel time $t^{\mathrm{free}}_{i^{\mathrm{L}}}$ (s). This can be written as a constraint on the cumulative outflow given as:

$$N^{\mathrm{out}}_{i^{\mathrm{L}}}(k^{\mathrm{c}}+1) \leq \gamma^{\mathrm{c,free}}_{i^{\mathrm{L}}} N^{\mathrm{in}}_{i^{\mathrm{L}}}(k^{\mathrm{c}} - k^{\mathrm{c,free}}_{i^{\mathrm{L}}} + 2) + (1 - \gamma^{\mathrm{c,free}}_{i^{\mathrm{L}}}) N^{\mathrm{in}}_{i^{\mathrm{L}}}(k^{\mathrm{c}} - k^{\mathrm{c,free}}_{i^{\mathrm{L}}} + 1)\,. \tag{6.3}$$

In (6.3) the number of time steps $k^{\mathrm{c,free}}_{i^{\mathrm{L}}} = \lceil t^{\mathrm{free}}_{i^{\mathrm{L}}}/T^{\mathrm{c}} \rceil$ (-), and the fraction $\gamma^{\mathrm{c,free}}_{i^{\mathrm{L}}} = k^{\mathrm{c,free}}_{i^{\mathrm{L}}} - t^{\mathrm{free}}_{i^{\mathrm{L}}}/T^{\mathrm{c}}$ (-) are used to linearly interpolate the cumulative curve as detailed in

[van de Weg et al., 2016]. The mathematical operator $\lceil \cdot \rceil$ rounds the argument of the function to the nearest integer that it higher than the argument of the function. In order to satisfy CFL conditions it should hold that $k_{i^{\mathrm{L}}}^{\mathrm{c,free}} \geq 2$.

Similarly, upstream propagating waves caused by spillback are included by bounding the cumulative link inflow from above so that a vehicle can only enter a link $t_{i^{\mathrm{L}}}^{\mathrm{shock}}$ (s) seconds after the vehicle $n_{i^{\mathrm{L}}}^{\max}$ (veh) has exited the link given as:

$$\begin{aligned} N_{i^{\mathrm{L}}}^{\mathrm{in}}(k^{\mathrm{c}}+1) \leq \gamma_{i^{\mathrm{L}}}^{\mathrm{c,shock}} N_{i^{\mathrm{L}}}^{\mathrm{out}}(k^{\mathrm{c}} - k_{i^{\mathrm{L}}}^{\mathrm{c,shock}} + 2) + \dots \\ (1-\gamma_{i^{\mathrm{L}}}^{\mathrm{c,shock}}) N_{i^{\mathrm{L}}}^{\mathrm{out}}(k^{\mathrm{c}} - k_{i^{\mathrm{L}}}^{\mathrm{c,shock}} + 1) + n_{i^{\mathrm{L}}}^{\max}\,, \end{aligned} \tag{6.4}$$

with the number of time steps $k_{i^{\mathrm{L}}}^{\mathrm{c,shock}} = \lceil t_{i^{\mathrm{L}}}^{\mathrm{shock}}/T^{\mathrm{c}} \rceil$ (-), and the fraction $\gamma_{i^{\mathrm{L}}}^{\mathrm{c,shock}} = k_{i^{\mathrm{L}}}^{\mathrm{c,shock}} - t_{i^{\mathrm{L}}}^{\mathrm{shock}}/T^{\mathrm{c}}$ (-). It should hold that $k_{i^{\mathrm{L}}}^{\mathrm{c,shock}} \geq 2$ in order to guarantee CFL conditions.

Outflow limitations at the network are modeled as external disturbances – i.e., inputs that cannot be affected by the control signal. So, when a link is at an exit of the network, an extra constraint is added:

$$N_{i^{\mathrm{L}}}^{\mathrm{out}}(k^{\mathrm{c}}+1) \leq N_{i^{\mathrm{L}}}^{\mathrm{out}}(k^{\mathrm{c}}) + q_{i^{\mathrm{L}}}^{\mathrm{out,max}}(k^{\mathrm{c}}) T^{\mathrm{c}}\,, \tag{6.5}$$

where $q_{i^{\mathrm{L}}}^{\mathrm{out,max}}(k^{\mathrm{c}})$ (veh/h) is the maximum outflow that can exit the link at time step k^{c}.

Origins are modeled as vertical queues via the following state update equations and constraints:

$$N_{i^{\mathrm{O}}}^{\mathrm{O,in}}(k^{\mathrm{c}}+1) = N_{i^{\mathrm{O}}}^{\mathrm{O,in}}(k^{\mathrm{c}}) + d_{i^{\mathrm{O}}}^{\mathrm{in}}(k^{\mathrm{c}}) T^{\mathrm{c}}\,, \tag{6.6}$$

$$N_{i^{\mathrm{O}}}^{\mathrm{O,out}}(k^{\mathrm{c}}+1) = N_{i^{\mathrm{O}}}^{\mathrm{O,out}}(k^{\mathrm{c}}) + q_{i^{\mathrm{O}}}^{\mathrm{cap}} T^{\mathrm{c}} b_{i^{\mathrm{O}}}^{\mathrm{eff,O}}(k^{\mathrm{c}})\,, \tag{6.7}$$

$$N_{i^{\mathrm{O}}}^{\mathrm{O,out}}(k^{\mathrm{c}}+1) \leq N_{i^{\mathrm{O}}}^{\mathrm{O,in}}(k^{\mathrm{c}}+1)\,. \tag{6.8}$$

with $q_{i^{\mathrm{O}}}^{\mathrm{cap}}$ (veh/h) the origin capacity.

The final constraints concern the effective fractions $b_{i^{\mathrm{L}}}^{\mathrm{eff}}(k^{\mathrm{c}})$ and $b_{i^{\mathrm{O}}}^{\mathrm{eff,O}}(k^{\mathrm{c}})$ of green-time which should be between 0 and 1. Additionally, if there is a conflict between links at an intersection – i.e., $\{j^{\mathrm{L}}, i^{\mathrm{L}}\} \in \mathcal{I}_{i^{\mathrm{con}}}^{\mathrm{conflict}}$ – the sum of the effective green fractions $b_{i^{\mathrm{L}}}^{\mathrm{eff}}(k^{\mathrm{c}}) + b_{j^{\mathrm{L}}}^{\mathrm{eff}}(k^{\mathrm{c}})$ should be less than $1 - \theta_{i^{\mathrm{con}}}$. The tuning parameter $\theta_{i^{\mathrm{con}}}$ (-) is used to prevent infeasible reference outflows which can occur when a clearance time has to be respected when switching link i^{L} to j^{L}. This results in the following constraints:

$$0 \leq b_{i^{\mathrm{L}}}^{\mathrm{eff}}(k^{\mathrm{c}}) \leq 1\,, \tag{6.9}$$

$$0 \leq b_{i^{\mathrm{O}}}^{\mathrm{eff,O}}(k^{\mathrm{c}}) \leq 1\,, \tag{6.10}$$

$$0 \leq b_{i^{\mathrm{L}}}^{\mathrm{eff}}(k^{\mathrm{c}}) + b_{j^{\mathrm{L}}}^{\mathrm{eff}}(k^{\mathrm{c}}) \leq 1 - \theta_{i^{\mathrm{con}}}\,. \tag{6.11}$$

The optimization problem

The objective of the linear optimization problem is to minimize the total time spent (TTS) J^{TTS} (veh·h) used by all the vehicles in the network over a prediction horizon

N^{p} (-) subject to the linear model and constraints presented in the previous section. The TTS can be expressed as the total number of vehicles in the network at every time step k^{c} multiplied with the sampling time T^{c} and summed over the time steps $k^{\mathrm{c}} = (k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+1,\ldots,(k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+N^{\mathrm{p}}+1$ given as:

$$J^{\mathrm{TTS}} = \sum_{k^{\mathrm{c}}=(k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+1}^{(k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+N^{\mathrm{p}}+1} T^{\mathrm{c}}\Bigg\{ \sum_{i^{\mathrm{L}}\in\mathcal{I}^{\mathrm{L}}} \left(N_{i^{\mathrm{L}}}^{\mathrm{in}}(k^{\mathrm{c}}) - N_{i^{\mathrm{L}}}^{\mathrm{out}}(k^{\mathrm{c}}) \right) + \ldots \sum_{i^{\mathrm{O}}\in\mathcal{I}^{\mathrm{O}}} \left(N_{i^{\mathrm{O}}}^{\mathrm{O,in}}(k^{\mathrm{c}}) - N_{i^{\mathrm{O}}}^{\mathrm{O,out}}(k^{\mathrm{c}}) \right) \Bigg\}, \tag{6.12}$$

Here, $\mathcal{I}^{\mathrm{L}}$ (-) represents the set of all links and $\mathcal{I}^{\mathrm{O}}$ (-) represents the set of all origins.

As in [van de Weg et al., 2016], minimizing the TTS can be written as the following linear optimization problem:

$$\min_{\bar{u}(k^{\mathrm{ref}})} Z\tilde{B}\bar{u}(k^{\mathrm{ref}}) + Z(\tilde{A}x(k^{\mathrm{ref}}) + \tilde{C}\bar{d}(k^{\mathrm{ref}}))\,, \tag{6.13}$$
$$\text{Subject to } M^{\mathrm{ineq}}\bar{u}(k^{\mathrm{ref}}) \leq V^{\mathrm{ineq}}\,,$$

Here, the matrices $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ as detailed in [van de Weg et al., 2016] describe the traffic dynamics, so that a prediction of the traffic state $\bar{x}(k^{\mathrm{ref}})$, as defined by equations 6.1, 6.2, 6.6, and 6.7, can be computed by multiplication of the control vector $\bar{u}(k^{\mathrm{ref}})$ by $\tilde{B}$, the initial traffic state $x(k^{\mathrm{ref}})$ by $\tilde{A}$, and a prediction of the disturbances $\bar{d}(k^{\mathrm{ref}})$ – i.e., inputs that cannot be controlled – by $\tilde{C}$. The matrix M^{ineq} and vector V^{ineq} as detailed in [van de Weg et al., 2016] contain the inequality constraints of equations 6.3, 6.4, 6.5, 6.8, 6.9, 6.10, and 6.11. Multiplication of the vector Z by the predicted state gives the TTS.

The vector $\bar{u}(k^{\mathrm{ref}})$ contains the effective fractions of green time $b_{i^{\mathrm{L}}}^{\mathrm{eff}}(k^{\mathrm{c}})$ and $b_{i^{\mathrm{O}}}^{\mathrm{eff}}(k^{\mathrm{c}})$ used by the links and origins in the network at the time steps $k^{\mathrm{c}} = (k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+1,\ldots,(k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+N^{\mathrm{p}}$:

$$\bar{u}(k^{\mathrm{ref}}) = \begin{bmatrix} u((k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+1) \\ \vdots \\ u((k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+N^{\mathrm{p}}) \end{bmatrix}. \tag{6.14}$$

The disturbance vector $\bar{d}(k^{\mathrm{ref}})$ contains the traffic demands $d(k^{\mathrm{c}})$ at time steps $k^{\mathrm{c}} = (k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+1,\ldots,(k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+N^{\mathrm{p}}$:

$$\bar{d}(k^{\mathrm{ref}}) = \begin{bmatrix} d((k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+1) \\ \vdots \\ d((k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+N^{\mathrm{p}}) \end{bmatrix}. \tag{6.15}$$

The control vector $u(k^{\mathrm{c}})$ and disturbance vector $d(k^{\mathrm{c}})$ at a time step k^{c} are defined as follows:

$$u(k^{\mathrm{c}}) = \begin{bmatrix} b_1^{\mathrm{eff}}(k^{\mathrm{c}}) & \ldots & b_{n^{\mathrm{L}}}^{\mathrm{eff}}(k^{\mathrm{c}}) & b_1^{\mathrm{eff,O}}(k^{\mathrm{c}}) & \ldots & b_{n^{\mathrm{O}}}^{\mathrm{eff,O}}(k^{\mathrm{c}}) \end{bmatrix}^{\top}, \tag{6.16}$$

$$d(k^{\mathrm{c}}) = \begin{bmatrix} d_1^{\mathrm{in}}(k^{\mathrm{c}}) & \ldots & d_{n^{\mathrm{O}}}^{\mathrm{in}}(k^{\mathrm{c}}) \end{bmatrix}^{\top}, \tag{6.17}$$

where n^{L} (-) indicates the number of links and n^{O} (-) the number of origins.

The reference trajectory

The outcome of the optimization problem (6.13) is the vector $\bar{u}^*(k^{\mathrm{ref}})$ (-). As noted before, this signal cannot be directly applied to the local intersection controllers due to the aggregated nature of the traffic flow model that is used to formulate the linear optimization problem. Instead, a reference trajectory is derived from the optimized signal $\bar{u}^*(k^{\mathrm{ref}})$.

A prediction of the traffic states $\bar{x}(k^{\mathrm{ref}})$ can be obtained as follows:

$$\bar{x}(k^{\mathrm{ref}}) = \tilde{A}x(k^{\mathrm{ref}}) + \tilde{B}\bar{u}^*(k^{\mathrm{ref}}) + \tilde{C}\bar{d}(k^{\mathrm{ref}})\,. \tag{6.18}$$

The prediction of the state $\bar{x}(k^{\mathrm{ref}})$ consists of the traffic states $x(k^{\mathrm{c}})$ at time steps $k^{\mathrm{c}} = (k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+2,\ldots,(k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+N^{\mathrm{p}}$ is given as:

$$\bar{x} = \begin{bmatrix} x((k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+2) & \ldots & x((k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+N^{\mathrm{p}}+1) \end{bmatrix}^{\top}\,. \tag{6.19}$$

In its turn, the state $x(k^{\mathrm{c}})$ consists of the states of the links $x^{\mathrm{L}}_{i^{\mathrm{L}}}(k^{\mathrm{c}})$ and origins $x^{\mathrm{L}}_{i^{\mathrm{O}}}(k^{\mathrm{c}})$ at time step k^{c}:

$$x(k^{\mathrm{c}}) = \begin{bmatrix} x^{\mathrm{L}}_1(k^{\mathrm{c}}) & \ldots & x^{\mathrm{L}}_{n^{\mathrm{L}}}(k^{\mathrm{c}}) & x^{\mathrm{O}}_1(k^{\mathrm{c}}) & \ldots & x^{\mathrm{O}}_{n^{\mathrm{O}}}(k^{\mathrm{c}}) \end{bmatrix}^{\top}\,. \tag{6.20}$$

The state of link $x^{\mathrm{L}}_{i^{\mathrm{L}}}(k^{\mathrm{c}})$ and origin $x^{\mathrm{O}}_{i^{\mathrm{O}}}(k^{\mathrm{c}})$ a time step k^{c} are defined as follows:

$$x^{\mathrm{L}}_{i^{\mathrm{L}}}(k^{\mathrm{c}}) = \begin{bmatrix} N^{\mathrm{out}}_{i^{\mathrm{L}}}(k^{\mathrm{c}}) & \ldots & N^{\mathrm{out}}_{i^{\mathrm{L}}}(k^{\mathrm{c}}-k^{\mathrm{c,shock}}_{i^{\mathrm{L}}}) & N^{\mathrm{in}}_{i^{\mathrm{L}}}(k^{\mathrm{c}}) & \ldots & N^{\mathrm{in}}_{i^{\mathrm{L}}}(k^{\mathrm{c}}-k^{\mathrm{c,free}}_{i^{\mathrm{L}}}) \end{bmatrix}^{\top}\,. \tag{6.21}$$

$$x^{\mathrm{O}}_{i^{\mathrm{O}}}(k^{\mathrm{c}}) = \begin{bmatrix} N^{\mathrm{O,out}}_{i^{\mathrm{O}}}(k^{\mathrm{c}}) & N^{\mathrm{O,in}}_{i^{\mathrm{O}}}(k^{\mathrm{c}}) \end{bmatrix}^{\top}\,. \tag{6.22}$$

Now, a reference cumulative outflow trajectory $N^{\mathrm{out,ref}}_{i^{\mathrm{L}}}(k^{\mathrm{c}})$, defined as:

$$N^{\mathrm{out,ref}}_{i^{\mathrm{L}}}(k^{\mathrm{ref}}) = \begin{bmatrix} N^{\mathrm{out}}_{i^{\mathrm{L}}}((k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+1 \\ N^{\mathrm{out}}_{i^{\mathrm{L}}}((k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+2 \\ \vdots \\ N^{\mathrm{out}}_{i^{\mathrm{L}}}((k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+N^{\mathrm{p}}+1) \end{bmatrix}\,, \tag{6.23}$$

can be derived from $\bar{x}(k^{\mathrm{c}})$ for every link $i^{\mathrm{L}} \in \mathcal{I}^{\mathrm{controlled}}$ for all the time steps where $k^{\mathrm{c}} = (k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+1,\ldots,(k^{\mathrm{ref}}-1)\epsilon^{\mathrm{c,ref}}+N^{\mathrm{p}}$.

Since the sampling time of the prediction model is a multiple of the measurements sampling time – i.e $T^{\mathrm{c}} = \epsilon^{\mathrm{c}}T$ –, the signal $N^{\mathrm{out,ref}}_{i^{\mathrm{L}}}(k^{\mathrm{ref}})$ has to be resampled. The reference outflow $\hat{N}^{\mathrm{out,ref}}_{i^{\mathrm{L}}}(\hat{k})$ at an arbitrary time step $\hat{k} \in (k^{\mathrm{ref}}-1)\epsilon^{\mathrm{ref}}+1,\ldots,(k^{\mathrm{ref}}+N^{\mathrm{p}}\epsilon^{\mathrm{c,ref}})\epsilon^{\mathrm{ref}}+1$ is given as follows:

$$\hat{N}^{\mathrm{out,ref}}_{i^{\mathrm{L}}}(\hat{k}) = (1-\gamma^{\mathrm{ref}}(\hat{k}))N^{\mathrm{out,ref}}_{i^{\mathrm{L}}}(\hat{k}^{\mathrm{c}}(\hat{k})) + \gamma^{\mathrm{ref}}(\hat{k}N^{\mathrm{out,ref}}_{i^{\mathrm{L}}}(\hat{k}^{\mathrm{c}}(\hat{k})+1)\,. \tag{6.24}$$

Here, the time step $\hat{k}^{\mathrm{c}}(\hat{k})$ is given as:

$$\hat{k}^{\mathrm{c}}(\hat{k}) = \lfloor \hat{k}/T^{\mathrm{c}} \rfloor\,, \tag{6.25}$$

and the fraction $\gamma^{\mathrm{ref}}(\hat{k})$ is the residual of a time step that $\hat{k}$ exceeds $\hat{k}^{\mathrm{c}}(\hat{k})$:

$$\gamma^{\mathrm{ref}}(\hat{k}) = \frac{\hat{k}-\hat{k}^{\mathrm{c}}(\hat{k})}{T^{\mathrm{c}}}\,. \tag{6.26}$$

6.2.3 Local intersection layer: greedy reference tracking

The task of the local intersection layer is to actuate at every time step k^{local} and at every intersection the stage that leads to the smallest reference tracking error. The reference tracking error of a stage is defined as a measure of the error between the reference outflow trajectories and the potential outflows of the different streams at an intersection when actuating that stage.

The stage selection is done in a decentralized way, which is possible because the time step T^{local} is chosen to be short – i.e., in the range of several seconds –, and no fixed stage sequence is assumed. The tracking strategy is called greedy, since it selects the stage that minimizes the reference tracking error for a short time horizon T^{local}. An alternative would be to implement a strategy that minimizes the tracking error over a longer time horizon. However, this would require predicting the outflow of many different stage sequences, and it would require taking into account the impact of the selected stage sequences of upstream and downstream intersections as well, leading to a complex optimization problem.

The greedy policy is computed for every intersection separately by carrying out the following steps:

1. predict for every stage the potential cumulative outflow of every link in the intersection when actuating the stage (see Section 6.2.3);
2. compute for every stage the resulting reference tracking error (see Section 6.2.3);
3. actuate the stage that is expected to realize the smallest reference tracking error (see Section 6.2.3).

Potential cumulative outflow prediction

The first step is to predict, for every intersection i^{inter} and stage $p_{i^{\text{inter}}}(k^{\text{local}}) \in \mathcal{P}^{\text{stages}}_{i^{\text{inter}}}$, with $\mathcal{P}^{\text{stages}}_{i^{\text{inter}}}$ the set of stages at the intersection, the potential cumulative outflows $N^{\text{out,p}}_{i^{\text{L}}}(\hat{k}|k, p_{i^{\text{inter}}}(k^{\text{local}}))$ (veh) of the links $i^{\text{L}} \in \mathcal{I}^{\text{US}}_{i^{\text{inter}}}$ directly upstream of the intersection using:

$$N^{\text{out,p}}_{i^{\text{L}}}(\hat{k}+1|k, p_{i^{\text{inter}}}(k^{\text{local}})) = \min \left\{ \begin{array}{l} N^{\text{out,p}}_{i^{\text{L}}}(\hat{k}|k, p_{i^{\text{inter}}}(k^{\text{local}})) + q^{\text{sat}}_{i^{\text{L}}} T b_{i^{\text{L}}}(\hat{k}), \dots \\ N^{\text{out,free}}_{i^{\text{L}}}(\hat{k}+1), \dots \\ N^{\text{out,sp}}_{i^{\text{L}}}(\hat{k}+1) \end{array} \right\} \forall i^{\text{L}} \in \mathcal{I}^{\text{US}}_{i^{\text{inter}}}, \tag{6.27}$$

for the time steps $\hat{k} = k+1, \dots, k+\epsilon^{\text{local}}+1$. In this equation, the maximum link outflow $N^{\text{out,free}}_{i^{\text{L}}}(k+1)$ (veh) in freeflow conditions is computed using

$$N^{\text{out,free}}_{i^{\text{L}}}(k+1) = \gamma^{\text{free}}_{i^{\text{L}}} N^{\text{in}}_{i^{\text{L}}}(k - k^{\text{free}}_{i^{\text{L}}} + 2) + (1 - \gamma^{\text{free}}_{i^{\text{L}}}) N^{\text{in}}_{i^{\text{L}}}(k - k^{\text{free}}_{i^{\text{L}}} + 1)\,. \tag{6.28}$$

It is assumed that $T^{\text{local}} < t_{i^{\text{L}}}^{\text{free}} \forall i^{\text{L}} \in \mathcal{I}_{i^{\text{inter}}}^{\text{US}}$, so that the outflow $N_{i^{\text{L}}}^{\text{out,free}}(k)$ depends on historical control decisions at the upstream intersections only. The maximum possible cumulative outflow under spillback from a downstream link $j^{\text{L}} \in \mathcal{I}_{i^{\text{L}}}^{\text{DS}}$ is computed using

$$N_{i^{\text{L}}}^{\text{out,sp}}(k+1) = N_{i^{\text{L}}}^{\text{out,p}}(k) + \gamma_{j^{\text{L}}}^{\text{shock}} N_{j^{\text{L}}}^{\text{out}}(k - k_{j^{\text{L}}}^{\text{shock}} + 2) + \dots \\ (1 - \gamma_{j^{\text{L}}}^{\text{shock}}) N_{j^{\text{L}}}^{\text{out}}(k - k_{j^{\text{L}}}^{\text{shock}} + 1) + n_{j^{\text{L}}}^{\max} - N_{j^{\text{L}}}^{\text{in,p}}(k)\,. \tag{6.29}$$

It is assumed that $T^{\text{local}} < t_{i^{\text{L}}}^{\text{shock}} \forall i^{\text{L}} \in \mathcal{I}_{i^{\text{inter}}}^{\text{DS}}$, so that the maximum outflow $N_{i^{\text{L}}}^{\text{out,sp}}(k)$ depends on historical control decisions at the downstream intersections only.

The cumulative link inflows $N_{i^{\text{L}}}^{\text{in,p}}(\hat{k}|k, p_{i^{\text{inter}}}(k^{\text{local}}))$ (veh) of the links $\mathcal{I}_{i^{\text{inter}}}^{\text{DS}}$ directly downstream of the intersection when actuating the stage $p_{i^{\text{inter}}}(k^{\text{local}})$ for the time steps $\hat{k} = k+1, \dots, k+\epsilon^{\text{local}}+1$ are updated using:

$$N_{i^{\text{L}}}^{\text{in,p}}(\hat{k}+1|k, p_{i^{\text{inter}}}(k^{\text{local}})) = \sum_{j^{\text{L}} \in \mathcal{I}_{i^{\text{inter}}}^{\text{US}}} \eta_{j^{\text{L}},i^{\text{L}}}(\hat{k}) \big(N_{i^{\text{L}}}^{\text{out,p}}(\hat{k}+1|k, p_{i^{\text{inter}}}(k^{\text{local}})) \dots \\ - N_{i^{\text{L}}}^{\text{out,p}}(\hat{k}|k, p_{i^{\text{inter}}}(k^{\text{local}})) \big) \, \forall i^{\text{L}} \in \mathcal{I}_{i^{\text{inter}}}^{\text{DS}}\,. \tag{6.30}$$

When clearance times have to be respected when switching from stage $p_{i^{\text{inter}}}(k^{\text{local}}-1)$ to stage $p_{i^{\text{inter}}}(k^{\text{local}})$, the corresponding values of $b_{i^{\text{L}}}(\hat{k})$ in (6.27) are set to 0 for the first $T_{i^{\text{L}}}^{\text{clear}}$ seconds.

Reference tracking error

Now that the predictions of the link outflows are available when actuating the different stages, the expected reference tracking error $\bar{e}_{i^{\text{inter}}}(p_{i^{\text{inter}}}(k^{\text{local}}))$ can be computed using:

$$\bar{e}_{i^{\text{inter}}}(p_{i^{\text{inter}}}(k^{\text{local}})) = \gamma^{\text{e}} \hat{e}^{\text{a}}_{i^{\text{inter}}}(p_{i^{\text{inter}}}(k^{\text{local}})) + (1-\gamma^{\text{e}}) \hat{e}^{\text{b}}_{i^{\text{inter}}}(p_{i^{\text{inter}}}(k^{\text{local}}))\,. \tag{6.31}$$

It is defined as the weighted average of the error $\hat{e}^{\text{a}}_{i^{\text{inter}}}(p_{i^{\text{inter}}}(k^{\text{local}}))$ – which is the square of the area between the reference outflow and the predicted outflow – computed using:

$$\hat{e}^{\text{a}}_{i^{\text{inter}}}(p_{i^{\text{inter}}}(k^{\text{local}})) = \sum_{\hat{k}=k+2}^{k+\epsilon^{\text{local}}+1} \sum_{i^{\text{L}} \in \mathcal{I}_{i^{\text{inter}}}^{\text{US}}} \left(\hat{N}_{i^{\text{L}}}^{\text{out,ref}}(\hat{k}) - N_{i^{\text{L}}}^{\text{out,p}}(\hat{k}) \right)^2 . \tag{6.32}$$

and of the error $\hat{e}^{\text{b}}_{i^{\text{inter}}}(p_{i^{\text{inter}}}(k^{\text{local}}))$ – which is the error between the total intersection reference outflow and total predicted intersection outflow $\hat{e}^{\text{b}}_{i^{\text{inter}}}(p_{i^{\text{inter}}}(k^{\text{local}}))$ – computed using:

$$\hat{e}^{\text{b}}_{i^{\text{inter}}}(p_{i^{\text{inter}}}(k^{\text{local}})) = \sum_{\hat{k}=k+2}^{k+\epsilon^{\text{local}}+1} \left| \left(\sum_{i^{\text{L}} \in \mathcal{I}_{i^{\text{inter}}}^{\text{US}}} \hat{N}_{i^{\text{L}}}^{\text{out,ref}}(\hat{k}) - \sum_{i^{\text{L}} \in \mathcal{I}_{i^{\text{inter}}}^{\text{US}}} N_{i^{\text{L}}}^{\text{out,p}}(\hat{k}) \right) \right| . \tag{6.33}$$

The parameter γ^{e} is introduced to balance the current reference tracking costs and the final reference tracking costs.

Stage actuation

The final step is the actuation of the stage $p^*_{i^{\text{inter}}}(k^{\text{local}})$ that leads to the smallest expected reference tracking error using:

$$p^*_{i^{\text{inter}}}(k^{\text{local}}) = \arg \min_{p_{i^{\text{inter}}} \in \mathcal{P}^{\text{stages}}_{i^{\text{inter}}}} \bar{e}_{i^{\text{inter}}}(p_{i^{\text{inter}}}(k^{\text{local}}))\,. \tag{6.34}$$

Numerical example

To clarify the reference tracking approach we have included the following simple numerical example. Assume that we have a network consisting of two conflicting links that can realize a flow equal to the saturation rate of 1000 veh/h when given green. It is also assumed that $T^{\text{local}} = 5$ s, and that the reference outflows for time step 1 to 12 are computed by the network coordination layer as 600 and 300 veh/h respectively, as shown in Figure 6.3. The inter-stage clearance time when switching from stage 1 to 2 and vice versa is assumed to be 2 seconds. Assume that at every time step we can choose between actuating stage 1 – i.e., giving green to link 1 and red to link 2 – or actuating stage 2 – i.e., giving red to link 1 and green to link 2.

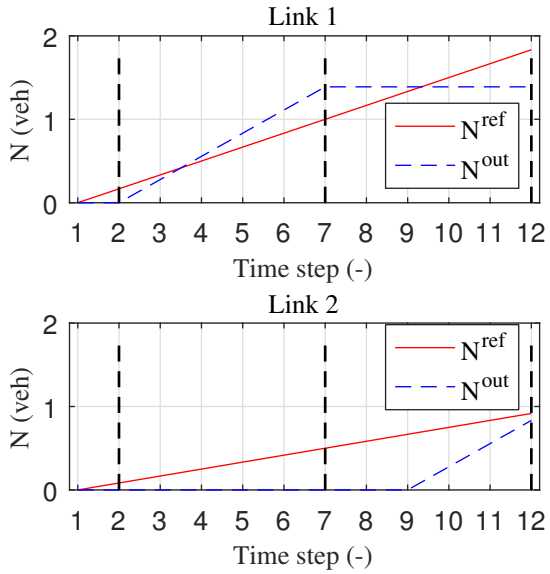


Figure 6.3: Small example of reference outflows and realized outflows.

At time step $k = 1$ the error is determined over time steps $k = 3$ to $k = 7$. For stage 1, the total error computed using (6.31) is 0.85 while the error for stage 2 is 1.82 given that γ^{e} = 0.3. Because the error of stage 1 is smaller it will be activated. Next, at time step $k = 6$, the error when actuating stage 1 is 2.28 while the error for actuating stage 2 is 1.82. Hence, stage 2 will be activated. Note that in the error calculation the inter-stage clearance time between stage 1 and stage 2 is accounted for.

6.3 Simulation experiments

Simulation experiments are carried out to show that the use of the individual intersection layer does not lead to significant performance degradation, and that the proposed framework is able to efficiently distribute the queues over the network in the presence of spillback. Additionally, the impact of the mismatch between the prediction and the process model is studied which is influenced by the selected process model and the chosen controller sampling times.

First simulations are carried out with the LTM as the process model, so that the mismatch between the process and prediction model is small. A comparison is made – in terms of TTS reduction and realized traffic states – with a controller that directly applies the reference outflows of the coordination layer to the model – which is only possible when using a macroscopic process model – giving the lowest possible TTS. This shows the TTS increase caused by the individual intersection layer. Next, the performance is compared with a greedy feedback policy that optimizes the signal timings of the local intersections. This provides insight into the ability of the proposed framework to distribute queues more efficiently over the network in the presence of spillback. Next, the microscopic model Vissim 5.30 is used as the process model, which introduces a larger mismatch.

In both simulations, the controller sampling times T^{local} and T^{ref} are varied and the impact on the TTS and reference tracking error is analyzed. It is expected that a smaller sampling time T^{local} leads to a lower TTS and a lower reference tracking error, because it allows more frequent switching of the stages. Similarly, it is expected that choosing a smaller sampling time T^{ref} reduces the reference tracking error but does not necessarily reduce the TTS.

6.3.1 Simulation set-up

The simulation set-up is shown in Figure 6.1. Every second, measurements are obtained from the process model – i.e., the LTM in Section 6.3.2, and Vissim in Section 6.3.3. The local control layer is updated every T^{local} seconds and the network coordination layer updates the reference trajectories every T^{ref} seconds. Figure 6.2 shows the network used in the simulations. It consists of three intersections; (1) top left, (2) top right, and (3) bottom right. The link lengths are indicated in the figure, where it must be noted that link 16 is 800 meters. It can also be seen that a bottleneck is located at the downstream end of link 7. This bottleneck is used to mimic a situation where downstream of the controlled network congestion is spilling back towards the controlled network. Alternatively, the bottleneck can represent a situation where the controlled network outflow is limited by a perimeter control strategy. A simulation period of 2500 seconds is considered. The demand pattern that is applied to the network consists of a high demand for the first 1800 seconds of respectively 900, 1100,

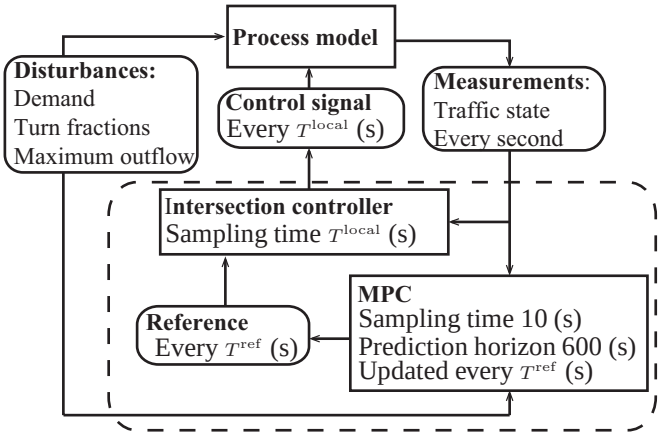


Figure 6.1: Schematic overview of the simulation set-up.

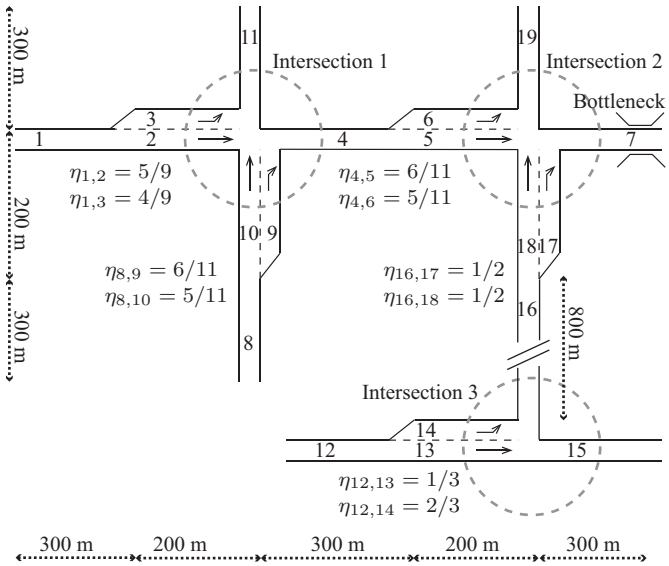


Figure 6.2: Schematic overview of the network used for the simulations, including the link lengths, location of the bottlenecks, and the turn-fractions.

and 1800 veh/h at links 1, 8, and 12. From time 1800 to 2500 seconds the demand is decreased to respectively 300, 250, and 200 veh/h at links 1, 8, and 12. This implies that in the high demand situation 600 veh/h want to go from links 5 to 7 and links 17 to 18, 500 veh/h from link 6 to link 19, and 600 veh/h from link 18 to link 19. The bottleneck at link 7 is activated from time 100 seconds with a capacity of 600 veh/h.

It is assumed that no measurement noise is present and that there is no uncertainty in the disturbance predictions. In this way, controlled experiments can be carried out that allow studying the controller behavior in detail. It must be noted that there is a mismatch between the process model and the prediction model caused by the difference in the local control signals and the MPC output.

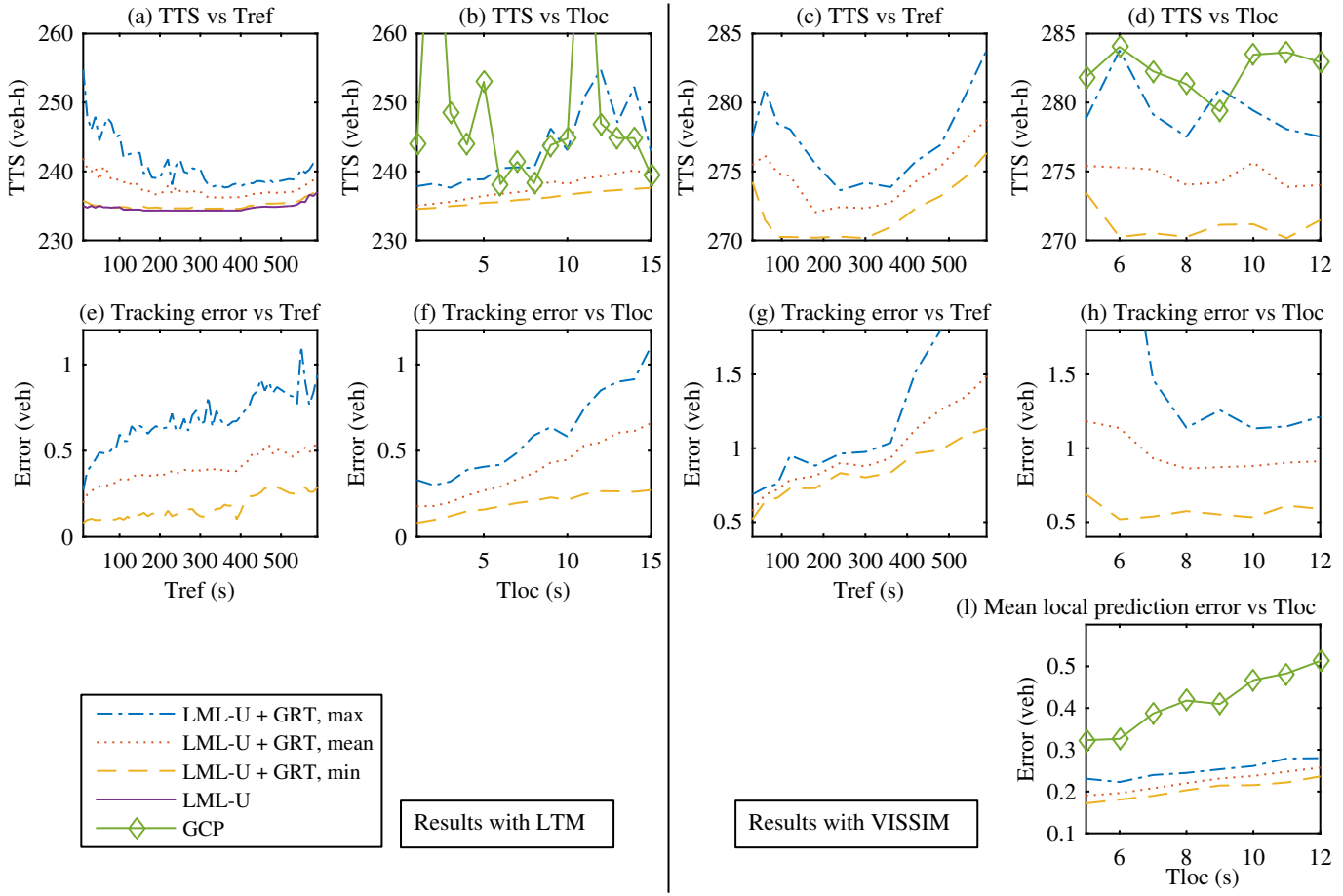


Figure 6.3: Simulation results for different set-ups. The two left columns represent the results obtained with the LTM, the two right columns represent results obtained with Vissim. The first row shows the impact of the controller sampling times T^{ref} and T^{local} on the TTS. The second row shows the impact of the sampling times on the mean reference tracking error. Plot (l) shows the impact of the sampling time T^{local} on the mean local prediction error. This result is not shown for the LTM because the prediction error is negligible, since the process and prediction models are identical. The max, mean, and min lines indicate the maximum, mean, and minimum realized TTS of the non-shown parameter (e.g. T^{local} in plot (a)).

6.3.2 Simulation set 1: macroscopic simulation using the LTM

The first set of evaluations is carried out using the LTM as the process model. These evaluations are carried out in order to gain insight into the quantitative controller performance. The LTM is used, because it enables a direct implementation of the reference outflows obtained from the network coordination layer and thus allows studying the reference tracking error incurred in the individual intersection control layer. The mean reference tracking error is defined as the average of the absolute difference between the reference outflows computed with the network coordination layer and the realized outflows.

Simulation set 1: set-up

The LTM is implemented as the process model with a sampling time step of 1 second. Clearance times are not considered in this simulation set, and the tuning parameters $\theta_{i^{\mathrm{con}}}$ are set to 0. This implies that the control strategies can actuate any stage at any time step T^{local}.

Three different control strategies are compared:

1. **LML-U + GRT**: this is the control strategy proposed in this paper.

2. **LML-U**: this is the LML-U strategy of the top layer with the optimized green-fractions directly applied to the network. Note that this implementation is not deployable, since these green-fractions can be simultaneously nonzero for conflicting directions in a time interval. Comparing with this control policy gives an idea of the best possible TTS that can be obtained.

3. **GCP**: this is a greedy control policy (GCP) that tries to actuate the stage at every time step T^{local} that will maximize the throughput of every individual intersection. This is realized by predicting for every stage the potential intersection outflow using the approach detailed in Section 6.2.3 and actuating the stage that will lead to the highest outflow. A comparison with this algorithm provides insight into the added value of the network coordination layer of the LML-U + GRT policy.

In the various simulations, the local control strategy sampling time T^{local} is varied from 1 to 15 seconds. The coordination layer sampling time T^{ref} is varied from 10 to 590 seconds. In this way the impact of the controller parameters on the controller performance can be studied. The prediction model used in the coordination layer uses a sampling time step of 10 seconds and a prediction horizon of 600 seconds. The factor γ^{e} is set to 0.3.

Simulation set 1: results

Several simulations were carried out with the different control strategies. The quantitative results are presented in the left two columns of Figure 6.3. First, the impact of changing the controller timings T^{ref} and T^{local} on the different controllers is discussed. After that, the performance of the different controllers is compared.

Figure 6.3 (a) and (e) show the impact of T^{ref} on the TTS and on the mean reference tracking error. For every sampling time T^{ref} there are multiple results, since the simulations were repeated for different values of T^{local}. Figure 6.3 (a) shows the impact of the coordination layer sampling time on the TTS. It can be observed from this figure that for low sampling times the TTS fluctuates considerably. When T^{ref} increases the fluctuations decrease, and for higher values of T^{ref} the TTS starts increasing again, which is mainly caused by the time T^{ref} being close to the prediction horizon of 600 seconds. Figure 6.3 (e) shows the impact of the sampling time T^{ref} on the mean reference tracking error. This plot shows a slight increase in the reference tracking error when increasing the time T^{ref}, although this result does not seem to be significant.

Figure 6.3 (b) and (f) show the impact of T^{local} on the TTS and on the mean reference tracking error. Figure 6.3 (b) shows that an increase in T^{local} results in an increase in the TTS. Similarly, Figure 6.3 (f) shows that an increase in T^{local} results in an increase in the reference tracking error. These results are best explained by the fact that a smaller sampling time T^{local} results in the possibility of more rapid stage switching, which allows for better tracking of the reference outflow trajectories.

Figure 6.3 (a) and (b) also show the realized TTS of the LML-U and GCP strategies. Figure 6.3 (a) shows that the LML-U strategy can realize the lowest TTS. It also shows that it is not sensitive to changes in the time T^{ref} until approximately 400 seconds. After that, the TTS increases due to the time T^{ref} getting close to the prediction horizon. The lowest TTS realized with the LML-U strategy is 234.33 veh·h . Figure 6.3 (b) shows that the TTS increases when increasing the sampling time T^{local}. The best performance realized by the LML-U + GRT strategy is 234.56 veh·h for T^{local} being 1 second. When setting T^{local} to a more realistic value of 5 seconds, the lowest TTS is 235.45 veh·h . In the case of the GCP, the lowest TTS realized is 238.16 veh·h .

These evaluations show that a sampling time T^{ref} in the range of 300 to 400 seconds is preferred for the performance. However, ideally T^{ref} is chosen small, so that the control strategy can quickly respond to disturbances. In order to reduce the sampling time T^{ref}, it is suggested to study the use of an observer in future research. The evaluations also show that the performance loss incurred by the switching of the stages is limited when the mismatch between the process and prediction model is small. Additionally, it is shown that a smaller local sampling time T^{local} results in better performance due to the ability to track the reference outflows more accurately.

6.3.3 Simulation set 2: microscopic simulation using Vissim

The second set of simulations is carried out with a microscopic simulation model. This allows us to study the performance when applied to a more complex process model. The quantitative performance is studied by comparing the control strategy to two other control strategies and studying the impact of changes in the controller parameters. Additionally, the qualitative performance is studied.

Simulation set 2: set-up

In this simulation set, Vissim 5.30 is used as the traffic flow model, with a sampling time step of 0.2 seconds. Measurements are gathered and sent to Matlab R2016a every second. The rest of the set-up is similar to that discussed in Section 6.3.2.

The same network model as in Figure 6.2 is used. However, the parameters used in the prediction model are different when compared to the parameters discussed in Section 6.3.2. The link parameters are shown in Table 6.1. The origin capacities are estimated as q_1^{cap} =2000 veh/h, q_8^{cap} =2000 veh/h, q_{12}^{cap} =2000 veh/h.

In the various simulations, the local control strategy sampling time T^{local} was varied from 5 to 12 seconds. The coordination layer sampling time T^{ref} was given values of 30, 60, 90, 120, 180, 240, 300, 360, 420, 480, 540, and 590 seconds. In this way, the impact of the controller parameters on the controller performance can be studied. The prediction model used in the coordination layer uses a sampling time step of 10 seconds and a prediction horizon of 600 seconds. The factor γ^{e} was set to 0.3. The clearance time between two conflicting links was set to 2 seconds, and the parameters $\theta_{i^{\text{con}}}$ were set to $4.4 \cdot 10^{-2}$.

Simulation set 2: quantitative results

The quantitative results are presented in the right two columns of Figure 6.3. First, the impact of the controller sampling times T^{ref} and T^{local} is discussed. After that the performance is compared to the GCP.

Figure 6.3 (c) shows the impact of T^{ref} on the TTS. It can be observed that the TTS is lowest for sampling times T^{ref} in the range of 200 to 300 seconds. This is in accordance with the results obtained with the LTM. The reason is that the reference outflows are determined for average dynamics. When using small values of T^{ref}, the frequent updates of the MPC signal do not allow a good representation of the average dynamics. For high sampling times T^{ref}, the impact of the mismatch between the process and prediction model becomes larger, as is also shown in Figure 6.3 (g).

Table 6.1: Link parameters used in the prediction model.

Link	t^{free}(s)	t^{shock}(s)	$n^{\max}$(veh)	q^{sat}(veh/h)	Link	t^{free}(s)	t^{shock}(s)	$n^{\max}$(veh)	q^{sat}(veh/h)
1	21.0	60.0	45	1961.9	11	21.0	58.0	46	2048.3
2	14.0	60.0	30	1916.1	12	21.0	56.4	44	1994.4
3	14.0	46.6	30	2000.0	13	14.0	61.0	31	1979.2
4	21.0	68.0	45	2369.8	14	14.0	70.0	30	1998.3
5	14.0	70.0	30	2369.8	15	21.0	58.0	46	1935.3
6	14.0	39.0	30	1848.5	16	57.0	205.0	119	1914.9
7	21.0	92.0	46	2023.0	17	14.0	60.0	30	2262.5
8	21.0	63.2	45	2150.9	18	14.0	48.3	31	2195.1
9	14.0	60.0	30	2000.0	19	21.0	53.4	47	1937.3
10	14.0	55.0	30	2000.0					

Figure 6.3 (d) shows the impact of T^{local} on the TTS. It can be observed that there is no clear connection between the sampling time T^{local} and the TTS. When studying Figure 6.3 (h), it is also clear that there is no strong connection between the sampling time T^{local} and the reference tracking error. This is best explained by the mismatch between the LTM and Vissim when predicting the intersection outflows with a time horizon in the range of 10 seconds. Figure 6.3 (l) shows the impact of T^{local} on the prediction error of the bottom layer.

When examining the realized TTS in Figure 6.3 (d), it can be seen that the LML-U + GRT strategy can realize a TTS of 270.17 veh·h while the GCP can realize a TTS of 279.35 veh·h . The reason for this, as discussed in the next subsection, is that the approach proposed in this paper distributes the queues over the network better. Also, when studying Figure 6.3 (l) it can be observed that the mean local prediction error of the GCP is consistently higher. The reason for this is that the predictions in the intersection layer are especially off when queues spill back to upstream intersections. This affects the GCP more, because that strategy causes much more spillback.

Simulation set 2: qualitative results

Figure 6.4 shows the number of vehicles over time in several links for the two different control strategies – i.e., the LML-U + GRT in the left column, and the GCP in the right column. Figure 6.5 shows the outflows of the network exits over time. The simulation results with $T^{\text{local}} = 9$ seconds and $T^{\text{ref}} = 300$ are used for the comparison. The vertical lines are used to indicate the time instants 300, 460, 650, and 1800 seconds respectively. Below, the behavior is discussed using these figures.

- Figure 6.4 (a) and (b) show that from time 80 to 300 the flow into the bottleneck exceeds the bottleneck capacity and a queue starts building up in link 7. This occurs when using either of the two policies.
- Figure 6.4 (c) and (d) show that at time 300 (indicated with the first vertical line) the spillback reaches links 5 and 17 and both controllers try to store as much traffic in these links in order to prevent blocking links 6 and 18.
- Around time 460 (indicated with the second vertical line) spillback cannot be avoided any more. The LML-U + GRT controller reduces the outflow of link 5 so that queues built up in links 5, 4, 2, and 9. In contrast to that, the GCP controller gives green to both links 5 and 17. This causes spillback towards links 4 and 16, which causes blocking of links 6 and 18.
- Next, around time 650 (indicated with the third vertical line) the LML-U + GRT blocks the outflow from link 17 in order to prevent spillback to links 8 and 1. As shown in Figure 6.4 (c), the number of vehicles in link 5 decreases while the number of vehicles in link 17 increases. It is interesting to see that links

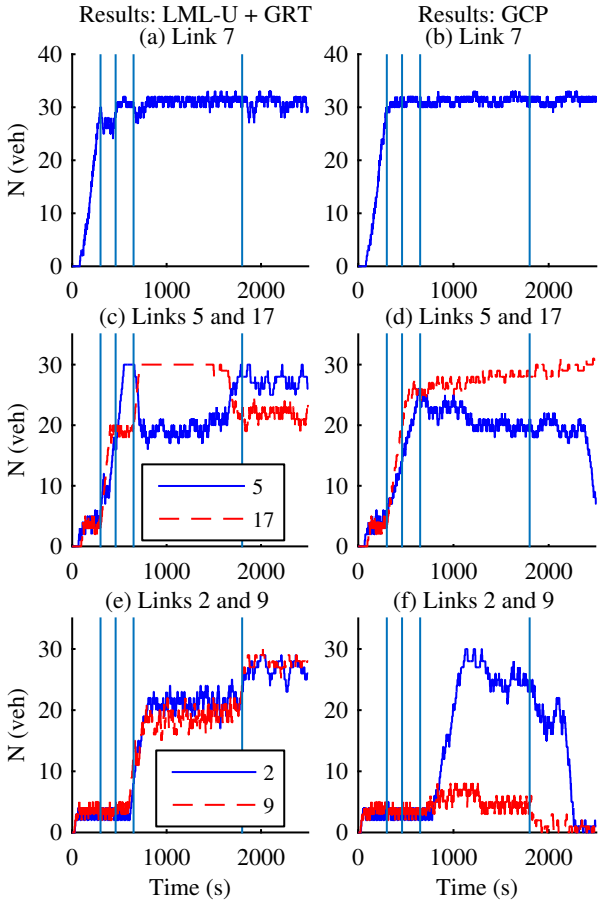


Figure 6.4: Number of vehicles in the links 7, 5, 17, 9, and 2 over time for the LML-U + GRT strategy in the left column and the GCP strategy in the right column. The vertical lines indicate the time instants 300, 460, 650, and 1800 seconds.

2 and 9 do not seem that full around time 650. This is due to the shock wave dynamics that cause a delay in the time when an outflow increase at link 5 leads to increased outflows at upstream links 2 and 9. Hence, only around time 800 seconds do the queues in links 2 and 9 become more or less stationary. The GCP controller does not have such a global view of the network, so the queue on link 2 grows, resulting in spillback to link 1 and an outflow reduction at link 11, as can be observed in Figure 6.5 (c).

- At time 1800 (indicated with the righter most vertical line) the demands decrease. Due to this, the outflow of link 5 can be reduced without triggering spillback to links 1 and 8 so that the queues on link 12, 14, 16, and 17 can be reduced.

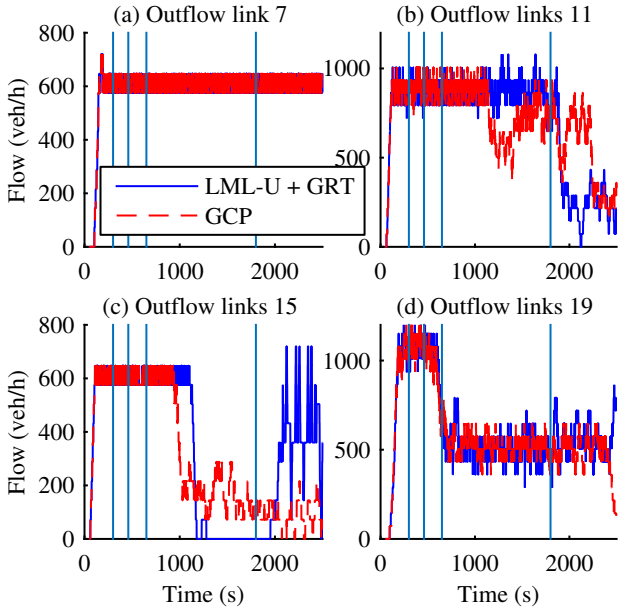


Figure 6.5: Outflow of links 7, 11, 15, and 19 over time for the LML-U + GRT strategy and the GCP strategy. The vertical lines indicate the time instants 300, 460, 650, and 1800 seconds.

6.4 Discussion

Several assumptions were made to simplify the problem addressed in this paper. This allowed us to combine optimization of the traffic flows at the network level with local signal controllers. This section discusses the implication of these assumptions and suggestions for relaxing them. It also discusses the scalability of the framework.

It was assumed that no minimum and maximum green times, no maximum or fixed cycle time, no off-set, and no fixed stage sequences had to be considered. Including these properties may affect the control performance, since, it reduces the control freedom. In order to correctly take these properties into account, the network coordination layer may need to be adjusted to reflect the impact of the different signal controller properties on the link outflows. Also, the logic of the local intersection control layer may need to be adopted to ensure that maximum green times, cycle times, and fixed stage sequences are realized. Depending on the problem type, this may be achieved by using heuristic approaches or optimization-based strategies. Hence, relaxing these assumptions may require some theoretical extensions and additional numerical evaluations.

Apart from that, an idealized set-up was assumed in which no noise and no uncertainties were considered, and in which normal vehicular traffic uses the network. The impact of uncertainties on the controller performance requires further investigation and, when needed, robust control strategies should be developed (e.g., see Tettamanti et al. [2014], Ukkusuri et al. [2010]). Different traffic types may be included by using a multi-modal LTM, and public transport priority may be included as constraints within the optimization problem.

The approach was designed for sub-networks consisting of (several) tens of intersections at maximum, and was tested on a small network consisting of three intersections. When applying the framework to larger networks, the computation time required by the network coordination layer increases. The size of the optimization vector is given as $(n^{\mathrm{L}}+n^{\mathrm{O}})N^{\mathrm{p}}$ (-) and the number of constraints is given as $(4n^{\mathrm{L}}+3n^{\mathrm{O}}+n^{\mathrm{E}}+n^{\mathrm{con}})N^{\mathrm{p}}$ (-), with n^{E} (-) the number of exits, and n^{con} (-) the number of conflicts between links.

6.5 Conclusions and recommendations

This paper proposes a hierarchical control framework for coordinated intersection control. The top layer – the network coordination layer – uses an efficient, linear MPC strategy for the optimization of network throughput. The output of the network coordination layer consists of reference outflow trajectories for the controlled links at intersections. The bottom layer consists of the individual intersection controllers that actuate the stage that minimizes the current reference tracking error. Simulations were carried out to test the impact of the controller timings and to compare the performance for the different timings. Simulations using the LTM as the process model indicated that the best performance can be obtained when using a moderate (around 200 to 300 seconds) sampling time for the network coordination layer. It was also shown that a smaller sampling time of the bottom layer leads to improved performance. It was found that the policy proposed in this paper can realize a TTS that is only 0.5% worse than the best possible performance when directly applying the signal of the network coordination layer. It was also shown that the controller can outperform a greedy control policy that tries to maximize the individual intersection throughput. Simulations using microscopic simulation revealed that the control strategy is capable of efficiently distributing the traffic over the network in spillback conditions, even when a large mismatch between the prediction and process model is present.

Further research can investigate the application of the framework to an intersection signal program where fixed stage sequences and minimum green times are included. Additionally, the application to a network that consists of heterogeneous vehicle types – e.g. vehicles, public transport, and bicycles – may be studied. Finally, further research can be carried out into the design of an observer so that the sampling time of the network coordination layer can be reduced.

Acknowledgments

This work is part of the research programme ‘The Application of Operations Research in Urban Transport’, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

This work was supported by the Australian Research Council (ARC) Future Fellowships FT120100723, and Discovery Project DP130100156 grants.

Chapter 7

Conclusion and recommendations

This dissertation addressed the challenge of developing efficient network-wide traffic control algorithms. The proposed algorithms are inspired by recent technological innovations and scientific insights as discussed in detail in Section 7.1. Because of the complexity of the traffic control problem for entire urban regions, this dissertation focused on developing control algorithms for medium-to-large scale networks consisting of tens of intersections or tens of kilometers of freeway. Section 7.2 presents, among others, recommendations for generalizing the results to entire urban regions and recommendations for further improving the proposed algorithms. Section 7.3 presents recommendations for applying the concepts in practice.

7.1 Summary and conclusions

This dissertation proposed *several computationally efficient network-wide traffic control algorithms for throughput improvement of medium-to-large scale freeway or urban traffic networks* that:

- coordinate the control actions of (different types of) actuators at different locations in the network,
- take the impact of the control actions on the network-wide performance over a time horizon into account.

Improving the network-wide throughput by coordinating the control actions of the actuators in a network is a complex problem. This complexity is caused by the large number of decision variables which is challenging from a computational point of view but it is also challenging from a theoretical point of view due to the many problem characteristics that need to be accounted for.

The proposed algorithms are designed to exploit recent technological innovations and scientific insights. The most relevant technological innovations for the algorithms proposed in this dissertation is the proliferation of in-vehicle technology enabling cooperative systems. This may provide more accurate traffic estimations by using floating car data (FCD), new data types, such as the planned route of individual vehicles, and more accurate actuation possibilities by considering the individual vehicle as the controlled element. The most relevant insight for the design of freeway traffic control algorithms in this dissertation is the application of shock wave theory to describe the effect of variable speed limits (VSLs) on the traffic flow. The most relevant insight for the design of urban traffic control algorithms in this dissertation is the application of the link transmission model (LTM) to describe the urban traffic dynamics.

This dissertation consists of two parts in which new algorithms are proposed based on these innovations and insights. The first part of this dissertation proposed two algorithms for the control of the speed of freeway traffic and to control on-ramp flows. The second part of this dissertation proposed three algorithms for the control of urban traffic networks using intersection control and route guidance.

Evaluations using simulation were carried out to study the ability of the algorithms to improve the balance between computation time and performance, and to study the qualitative behavior of the different controllers. In the remainder, we take a closer look at the specific conclusions per thesis chapter.

Chapter 2 proposed a *cooperative speed control algorithm to resolve jam waves* in order to improve the freeway throughput. The algorithm uses the individual vehicles as detectors and actuators assuming a 100% penetration rate and work as follows. The individual vehicles detect based on their (historical) speed data whether they are driving in a jam or not, this is called the detection mode. The vehicles send their detection mode, speed and position data to the roadside. The roadside system then uses this data to determine the location of the jam head and the required driving strategies – called driving modes – of the vehicles on different segments of the freeway. The roadside system then sends a generalized message to the vehicles indicating between which location which driving mode is active. Finally, vehicles adjust their speed accordingly either by following in-vehicle instructions, or by directly influencing the speed of the vehicle. This set-up was chosen, since, it does not require to store privacy sensitive GPS position and speed data at the roadside, nor is it required to address individual vehicles using a unique ID. Evaluations using simulation showed that the algorithm can improve the freeway throughput by resolving a jam wave, and can stabilize traffic. In the idealized case of a one-lane freeway this led to a total time spent (TTS) reduction of 7.3% and in a more realistic case consisting of a two-lane freeway this lead to an average TTS reduction of 17.3% compared to the no-control case. It was also shown that the algorithm can realize a similar qualitative behavior when compared to the SPECIALIST algorithm. This is an important observation, since the SPECIALIST algorithm has been proven in the field.

This chapter shows that an efficient algorithm for the control of traffic flows using co-

operative systems can be developed. An advantage of the algorithm is that it requires a negligible amount of computation time which is important, given the large amount of decision variables involved when controlling the speed of all the vehicles on a freeway. This chapter did not indicate whether the use of these technologies will also lead to a performance gain in practice when compared to existing infrastructure-based technologies. In order to conclude this, it is required to adapt the algorithm to be applicable to more realistic situations with lower penetration rates, and noisy measurement data, and to assess the performance using extensive simulation studies accordingly. The algorithm is designed to resolve a jam wave on a homogeneous stretch of freeway and is not specifically designed to account for overtaking. Hence, additional research is required to study the application of the algorithm to more general situations that include not only jam waves but also other congestion types, to study the impact of overtaking on the algorithm, and to study the application of the algorithm to more general road lay-outs that include, among others, merges, diverges, on-ramps, and off-ramps.

Chapter 3 proposed an *efficient optimization approach for integrated control of VSLs and ramp metering (RM)* to improve the freeway throughput. The balance between computation time and performance was improved by reducing the number of optimization variables through parameterization of the VSL and RM signal. The parameterized VSL signal consists of the speed with which the downstream and upstream boundaries of a speed-limited area propagate. It is assumed that the average speed inside the speed-limited area is equal to the effective speed of the imposed speed-limits. By changing the speeds of the downstream and upstream boundaries of the speed-limited area, the density and flow inside and downstream of it can be influenced. This parameterization reduces the number of variables from the number of variable message signs to just 2 per time-step. The number of RM control variables per RM installation is reduced from the number of controlled time-steps within the control horizon to 5. The first RM decision variable is the time when a feedback RM strategy based on the well-known ALINEA algorithm is switched on. The density set-point of this strategy is the second decision variable. The third variable is the time when the set-point is adjusted to a new set-point which is the fourth decision variable. The final decision variable is the time when RM is switched off. This parameterization reduces the number of decision variables while still being able to switch between various RM policies. Evaluations using macroscopic simulation indicated that a better balance between computation time and performance was realized for a VSL-only and an integrated VSL and RM set-up when compared to a nominal model predictive control (MPC) algorithm. It was also shown that the algorithm is capable of improving the throughput in two different traffic situations, namely, when resolving a jam wave, and when preventing bottleneck congestion. The algorithm was also analyzed qualitatively, showing the differences between using VSL-only, RM-only and integrated VSL and RM.

This chapter showed that an efficient optimization approach for integrated control of VSLs and RM can be developed. The proposed parameterization has several advantages. First, due to the reduction of the computation time it enables the use of more

complex prediction models or to control larger networks. Second, the imposed speed-limited area is more insightful when compared to a nominal MPC strategy which may help to obtain trust of the authorities in the proposed strategy. Two approaches to reduce the computation time even further may be to; 1) provide good starting points for the optimization problems – for instance, based on the SPECIALIST control scheme – or 2) by including an end-cost function that allows to reduce the prediction horizon. Additional research is required to apply the proposed strategy in practice. One important area of investigation is the quality of traffic state estimations and disturbance predictions and the influence of that data on the performance of the control algorithm. Another relevant direction for further research is the integration with cooperative systems. For instance, this may require to translate the optimized signals to a control signal for individual vehicles, but it may also be needed to accurately describe the impact of in-vehicle measures – such as, in-vehicle VSLs – in the (macroscopic) prediction model.

Chapter 4 proposed an *efficient MPC strategy for optimizing the traffic flows that cross intersections* in order to improve the urban road network throughput. The proposed MPC strategy uses the LTM as the prediction model and aggregates the traffic flow dynamics to tens of seconds so that, instead of green-times, the fractions of green-time used by every stream are the optimization variables, which are continuous. The use of the LTM as the prediction model has several advantages. First, the LTM describes the link dynamics using just two traffic states, namely, the cumulative inflow and outflow. This reduces the dimension of the corresponding optimization problem when compared to approaches that divide a link into segments. Second, the LTM is capable of describing downstream propagating waves caused by free flowing traffic, queuing dynamics, and upstream propagating waves caused by spillback. The downstream propagating waves allow to coordinate the flows exchanged between intersections in free flow conditions. The queuing dynamics and upstream propagating waves allow to distribute the queues over the network in congested conditions. It is shown using macroscopic simulation that the use of the LTM leads to a better balance between computation time and realized throughput when compared to a linear MPC strategy based on the cell transmission model. It is also shown that the inclusion of upstream propagating waves leads to better throughput when compared to a linear MPC strategy based on the store-and-forward model but also that this requires more computation time. The strategy was able to optimize the flows in a large network consisting of 96 controlled elements in a maximum CPU time just under 1 minute.

This chapter showed that it is possible to optimize the flows in an urban network using a computationally efficient algorithm that can consider downstream and upstream propagating waves and queuing dynamics. A major advantage when compared to other comparable approaches is that it can distribute queues more efficiently over the network in the oversaturated regime when considering aggregated traffic dynamics. However, it is not straightforward to conclude whether this approach will also help in practice. The reason for this is that this would require to translate the optimized flows to signal

timing plans. This is not a trivial problem because several degrees of freedom exist when designing a signal plan; e.g. 1) the cycle length of the intersections; 2) the offset between intersections; 3) the set of stages that are included in the signal plan; and 4) the order of the stages within each cycle. All these elements of the signal plan may influence the controller performance, and thus need to be chosen properly.

Chapter 5 proposed an *efficient optimization strategy for the control of flows that cross intersections and routing decisions* in order to improve the network throughput. The inclusion of routing decisions results in the optimization problem of Chapter 4 in a non-linear prediction model and optimization problem. Therefore, an efficient optimization algorithm of the sequential linear programming (SLP) type is used. Such an algorithm uses the gradient of the objective function in an operating point. Conventional solvers use a numerical approximation of the gradient which is very time consuming for large optimization problems. Therefore, an analytic approximation of the gradient is derived in this chapter. This gradient is obtained by predicting the traffic state using the non-linear model for a given control signal. Next, the turn-fractions are estimated from the predicted traffic states and a similar linear prediction model as used in Chapter 4 is obtained. After that, the impact of changing the flows and routing decisions onto the turn-fractions is estimated analytically and included in the linear model. Evaluations using macroscopic simulation revealed that the algorithm can realize a better balance between computation time and throughput when compared to applying a conventional numerical optimization algorithm.

This chapter showed that the intersection flows and routing decisions can be optimized using an efficient algorithm. The realized computation time gain is indeed promising, but additional research is required before applying the strategy in practice. Similarly as in the previous chapter, a translation of the optimized flows to signal timings is needed. Additionally, it has to be determined what the compliance to the routing decisions is and either the optimization approach has to consider compliance, or the actuation has to be adapted to ensure full compliance using for instance (monetary) incentives. Another direction for further research is to further extend the procedure to approximate the gradient.

Chapter 6 proposed a *hierarchical control framework for coordinating the signal timings of intersections* in order to improve the network throughput. The framework consists of two layers. The top layer uses the MPC strategy proposed in Chapter 4 to optimize the aggregated flows at intersections. These flows are send to the bottom layer as outflow references that consist of the individual intersection controllers. The task of the individual intersection controllers is to actuate at every controller time step of the bottom layer the stage – i.e., set of streams that can be active simultaneously – that is expected to minimize the reference tracking error during the next time-step. In this way, the top layer can use an efficient MPC strategy to distribute the average traffic flows over the network and the individual intersection control problem is simplified to a local problem that still leads to network-wide performance improvements. Evaluations using macroscopic simulation are carried out to study the added value of

the network coordination layer, and the impact of the timing onto the controller performance. Evaluations using microscopic simulation revealed the controllers ability to improve the throughput by distributing the queues over the network in a more realistic set-up when compared to independently maximizing the outflow of the individual intersections.

This chapter showed that coordinating the flows exchanged between intersections using the MPC approach proposed in Chapter 4 can improve the network performance, even when considering signal timings. It cannot be concluded whether this will also work well in practice for several reasons. First, the signal timing plan was simplified when compared to a realistic signal timing plan. The approach may need to be extended or modified in order to apply it to more realistic signal timing plans. Second, more extensive evaluations are required that include, among others, noise and uncertainties, more realistic networks, and more stochastics.

7.2 Recommendations for further research

The algorithms proposed in this dissertation mainly addressed the challenge of realizing a better trade-off between computation time and performance of network-wide traffic control algorithms. This section details several recommendations for further research. First, Section 7.2.1 presents additional research to integrate the proposed sub-network controllers in an approach for an entire urban region. Next, Section 7.2.2 presents recommendations to further improve the proposed algorithms.

7.2.1 Coordinated control of urban regions

The algorithms proposed in this dissertation were designed for *medium-to-large scale urban or freeway* networks. This simplified the sub-network control problem so that a good trade-off between computation time and performance could be realized. Additional research is required to study how the flows exchanged between different sub-networks have to be managed in order to improve the performance of an entire urban region which may span several hundreds of kilometers and houses several millions of people.

More research is needed into the development and application of efficient control algorithms that coordinate the flows exchanged between different sub-networks. Similarly as with the control algorithms proposed in this dissertation, a balance has to be found between computation time and performance. Such an algorithm would have two functions, namely, optimizing the flows exchanged between regions, and providing predictions of the inflows and desired outflows of the different sub-networks. The controller sampling time can be chosen relatively large, i.e., in the range of tens of minutes. Below two promising research directions are discussed.

One promising approach is the use of the NFD within an MPC framework as done by Hajiahmadi et al. [2013b], Haddad et al. [2012]. The advantage of this approach is that the complexity of the optimization problem can be reduced drastically. However, also several issues have to be investigated to analyze the feasibility of applying the framework. First, both the control strategies used in the sub-networks, and by the regional MPC strategy itself may affect the shape of the NFD. It may be investigated whether this affects the controller performance, and when needed it this may be included in the controller design. See Zhou et al. [2016] for an example of a hierarchical control framework using an MPC based on the NFD for the higher level and an MPC based on the S-model to control the sub-networks. Second, the NFD is designed for urban networks. However, also an efficient model for freeway networks is required that is capable of predicting the flows in a large freeway network using very little computation time. Third, an approach is needed to choose the sub-network boundaries so that optimal network performance can be achieved, see e.g. [Ji and Geroliminis, 2012, Ma et al., 2009, Etemadnia et al., 2014, Hisai and Usami, 2006].

Another promising approach is the use of a decentralized control framework based on distributed MPC. Frejo and Camacho [2012] have extensively studied the application of distributed MPC to a freeway network, Tettamanti and Varga [2010] proposed a distributed optimization algorithm for urban traffic control. In distributed MPC, every sub-network is optimizing an objective function that expresses the local performance and the impact of the realized outflows onto the rest of the network. One of the challenges of this framework is to find an expression for the impact of the sub-network control actions onto the entire network performance.

7.2.2 Further improvements of proposed algorithms

The proposed algorithms may be improved in various ways. Below, various directions are detailed that discuss additional research to further speed up the algorithms, to make even more use of in-vehicle technology enabling cooperative systems, to study the impact of noise and uncertainties, or to further generalize the proposed algorithms.

Further improvement of the balance between computation time and performance

There is room for further improving the balance between computation time and performance of the proposed algorithms. In fact, it is likely that there will always be new innovations and scientific insights to improve this balance. The directions discussed here are limited to extensions of the proposed algorithms.

The prediction horizon may be shortened by *including end-cost functions.* A reduction of the prediction horizon reduces the dimension of the optimization problem, and as a consequence reduces the computation time. However, this also increases the likelihood that the optimized signal is myopic, i.e., it minimizes the short term but not the

long term costs. Including end-cost functions is a promising direction to reduce the prediction horizon while being able to take the long-term costs into account. See e.g. Jamshidnejad et al. [2016] who proposed an MPC strategy for urban traffic network control that uses end-point penalties.

Another approach to reduce the computation time is to *develop a procedure to choose good starting points* for the optimization. This may especially help when multiple local minima exist. An example of such a starting point for the parameterized MPC strategy proposed in Chapter 3 may be the use of the SPECIALIST control scheme. The MPC can then further improve the shape of the imposed VSL area so that it optimizes for the details in the traffic flows, such as, on-ramp and off-ramp traffic. A risk of such a strategy is that the local optima found in this way are not close to the global optimum due to the selected control strategy.

Using in-vehicle technology enabling cooperative systems

In-vehicle technology enabling cooperative systems has the potential to improve detection and actuation possibilities which may lead to more efficient traffic control algorithms. This dissertation presented an algorithm in Chapter 2 that is specifically designed for an in-vehicle system. One of the issues of using this technology to control the traffic is that the number of actuators is very large. Hence, optimizing the desired behavior of every vehicle in order to improve the network throughput results in a very large optimization problem. An extension of the COSCAL v1 algorithm to account for on-ramp flows is presented in van de Weg et al. [2014a]. It is expected that extending the theory to more complex road lay-outs consisting of multiple on-ramps, off-ramps, and lane-drops may be difficult to realize using an analytic approach as proposed in van de Weg et al. [2014a].

A promising research direction is to develop a *multi-layer or hierarchical control strategy for cooperative systems*. The top layer optimizes the network performance based on flows while the lower level controls the individual vehicles to optimize the different segments. This would require to develop traffic flow models that provide a good balance between computational complexity and accuracy of modeling the in-vehicle system.

In-vehicle technology and road-side technology will co-exist in the coming decades. One of the reasons for this is that not all vehicles will be equipped with in-vehicle technology enabling cooperative systems. Therefore, a system that uses cooperative systems should be able to work with a mix of infrastructure-based systems and in-vehicle technology. Hence, it is recommended to develop algorithms that are able of utilizing both in-vehicle and infrastructure-based technology. Note that Mahajan et al. [2015] proposed an infrastructure-based variant of the COSCAL v1 algorithm called COSCAL v2.

Impact of noise and uncertainties

An ideal world was considered in this dissertation in which no measurement noise or disturbance prediction uncertainties were considered. Also, case studies were designed to study the balance between computation time and controller performance by using simulation models that provided a small mismatch between the process and prediction models. Additional research is required when relaxing these assumptions as explained below.

The controller performance when subject to noise and uncertainties has to be investigated. The first step would be to study the impact of measurement noise or uncertainties in the disturbance predictions or in the prediction models on the controller performance. It is well-known that the performance of control algorithms that anticipate the impact of local control action on the rest of the network may be affected by prediction uncertainties. Additional simulation studies are recommended in which the impact of these uncertainties is systematically investigated. Depending on the outcome of these evaluations, the following additional research may be required to limit the impact of noise and uncertainties.

Observers and filters may need to be designed to filter the measurement noise and estimate the traffic state that is used by the control algorithms. Since traffic state estimation is an intensively studied research area, off-the-shelve algorithms may be used. It is recommended to include observers and filters and evaluate the impact of noise and uncertainties on the controller performance using more realistic simulation experiments. Additionally, new data types, such as, FCD, radar, Bluetooth, or video data, may be used to acquire better estimates of the traffic state.

Disturbance prediction algorithms may have to be developed that provide the sub-network control algorithms with an estimate of the near-future disturbances, such as, the traffic demand, the turn-fractions or the route choice. Currently, such algorithms base their predictions on historical and real-time inductive detector loop data. However, the (planned) routes of road-users may drastically improve this estimation, although it may also cause some interesting control problems as discussed in Chapter 5. Additionally, the inclusion of a sub-network coordination algorithm may be used to control the flows which may also influence the available prediction of the disturbances.

Robust control algorithms may need to be applied that guarantee a certain performance in uncertain traffic situations. These algorithms may be designed to recover quickly when an incident occurs – such as a crash or a bridge opening – or to prevent performance degradation under uncertain conditions. A promising research direction is the use of robust MPC, see e.g. [Tettamanti et al., 2014]. However, a challenge is that accounting for uncertainty can be computationally complex. Jansen [2016] took a step in this direction by extending the linear MPC strategy proposed in Chapter 4 so that it keeps additional storage space in the links in the over-saturated traffic regime using a computationally efficient optimization algorithm.

Heterogeneous traffic

This dissertation focused on networks used by vehicular traffic. In practice, traffic networks are used by heterogeneous traffic consisting of, among others, 'standard' vehicles, trucks, bicycles, pedestrians, emergency vehicles, and public transport. Further research is recommended to study the application of the control algorithms proposed in this dissertation to a wider variety of traffic. One research direction is the use of robust control algorithms as discussed above. Another is the use of multi-mode traffic flow models.

Various objective functions

This dissertation focused on the improvement of throughput. In practice, other performance indicators have to be considered as well. Additional research is recommended to extend the proposed algorithms to include other objective functions. It may also be the case that the objective function may vary between the different sub-networks, since, the (different) authorities may have different objectives. This requires additional research as well.

Other types of actuators

This dissertation proposed algorithms for *variable speed limits, ramp metering, traffic lights, and route guidance.* Other types of actuators may also be included, especially considering the fact that the proliferation of in-vehicles technologies enabling cooperative systems allow more alternative types of traffic control measures. This requires first to study how these actuators affect the network flows, and subsequently to include them in an efficient control or optimization algorithm.

Integration with demand management measures

This dissertation focused on maximizing the use of the existing infrastructure. However, when there is simply too much traffic, inefficiencies will be inevitable. Hence, it is recommended to study the integration of traffic control measures with demand management. Especially the interaction between demand management and traffic control measures is an interesting research topic.

7.3 Towards application of concepts in practice

Several open questions need to be addressed before the proposed algorithms can be applied in practice. Some of these questions are related to studying practical issues,

such as the impact of noise and uncertainties on the performance of the algorithms. Other questions are related to preparing the algorithms for applications in practice. These questions are discussed below.

The impact of noise and uncertainties on the performance

As discussed above, noise and uncertainties affect the performance of the algorithms. It is recommended to first systematically assess the quality of the data available for traffic state estimation and demand predictions. This may provide insight into the certainty with which the demand can be predicted. Next, it can be assessed to what extend this realistic data affects the controller performance. Based on these assessments it may be recommended to develop better state estimation or demand prediction algorithms, install better detectors, or to adjust the algorithms by making them robust or tune more conservatively as also discussed in Section 7.2.2.

Assessment of the expected impact of control measures

Assessment of the expected impact of control measures is required to determine whether it is beneficial to implement a measure at a site. This requires to develop a framework to quantify the potential gain at a specific site. This is not a trivial task, since, it requires a good data set, but also requires theoretical developments. An example of such an analysis is presented in van de Weg et al. [2014b]. There, an ex-ante data analysis technique is proposed to assess the potential gain of applying a VSL measure to resolve jam waves on the A13 freeway in the Netherlands.

Satisfying practical requirements

The algorithms may need to be adapted to satisfy all the requirements that are needed for practical application. These requirements may be hardware related. For instance, the algorithms need to be able to deal with communication delays between detectors, the control system, and the actuators. Apart from that, the algorithms may need to be tailored to work on a specific site. For instance, the algorithm may need to be adopted to include a specific road lay-out, to account for priority vehicles, such as emergency vehicles or public transportation, and to account for various types of road-users, such as vehicles, pedestrians, and cyclist. Also, the compliance with the measures may need to be included in the algorithms, or new actuation approaches may need to be developed to guarantee a certain compliance rate.

Extensive evaluations

It is recommended to carry out extensive simulations of the proposed algorithms before implementation at a specific test site. These simulations can provide, among others,

insight into the potential performance gains of the proposed algorithms in practice, insight into the algorithmic behavior, and insight into the impact of changing the parameter values from which tuning guidelines for practical application can be derived. This requires to evaluate using simulations which are as close to the real set-up as possible, i.e., the road lay-out and traffic situations should be similar.

References

H. M. Abdul Aziz and S. Ukkusuri. Unified framework for dynamic traffic assignment and signal control with cell transmission model. *Transportation Research Record: Journal of the Transportation Research Board*, 2311(1):73–84, 2012.

K. Aboudolas, M. Papageorgiou, A. Kouvelas, and E. Kosmatopoulos. A rolling-horizon quadratic-programming approach to the signal control problem in large-scale congested urban road networks. *Transportation Research Part C: Emerging Technologies*, 18(5):680–694, 2010.

R. Akçelik and M. J. Maher. Route control of traffic in urban road networks: review and principles. *Transportation Research*, 11(1):15–24, 1977.

R. E. Allsop. Some possibilities for using traffic control to influence trip distribution and route choice. In *Transportation and Traffic Theory, Proceedings,* volume 6, 1974.

R. E. Allsop and J. A. Charlesworth. Traffic in a signal-controlled road network: An example of different signal timings including different routeing. *Traffic Engineering & Control*, 18(Analytic), 1977.

G. M. Arnaout and J.-P. Arnaout. Exploring the effects of cooperative adaptive cruise control on highway traffic flow using microscopic traffic simulation. *Transportation Planning and Technology*, 37(2):186–199, 2014.

A. M. Bayen and A. D. M. Patire. Mobile century: A traffic sensing field experiment using GPS mobile phones. Technical Report 15572269, California Center for Innovative Transportation, 2010.

A. Bemporad and M. Morari. Robust model predictive control: A survey. In *Robustness in identification and control*, pages 207–226. Springer, Berlin, Germany, 1999.

M. Burger, M. van den Berg, A. Hegyi, B. De Schutter, and J. Hellendoorn. Considerations for model-based traffic control. *Transportation Research Part C: Emerging Technologies*, 35:1–19, 2013.

R. C. Carlson, I. Papamichail, and M. Papageorgiou. Local feedback-based mainstream traffic flow control on motorways using variable speed limits. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1261–1276, 2011.

R. C. Carlson, I. Papamichail, and M. Papageorgiou. Integrated feedback ramp metering and mainstream traffic flow control on motorways using variable speed limits. *Transportation Research Part C: Emerging Technologies*, 46:209–221, 2014.

D. Chen, S. Ahn, and A. Hegyi. Variable speed limit control for steady and oscillatory queues at fixed freeway bottlenecks. *Transportation Research Part B: Methodological*, 70(0):340–358, 2014.

H.-K. Chen and C.-F. Hsueh. *Combining signal timing plan and dynamic traffic assignment*. Transportation Research Board, 1997.

O. J. Chen. *Integration of dynamic traffic control and assignment*. PhD thesis, Massachusetts Institute of Technology, 1998.

C. Daganzo. The cell transmission model, part II: network traffic. *Transportation Research Part B: Methodological*, 29(2):79–93, 1995.

C. Daganzo. Urban gridlock: macroscopic modeling and mitigation approaches. *Transportation Research Part B: Methodological*, 41(1):49–62, 2007.

C. Diakaki, V. Dinopoulou, K. Aboudolas, M. Papageorgiou, E. Ben-Shabat, E. Seider, and A. Leibov. Extensions and new applications of the traffic-responsive urban control strategy: Coordinated signal control for urban networks. *Transportation Research Record: Journal of the Transportation Research Board*, 1856(1):202–211, 2003.

H. Etemadnia, K. Abdelghany, and A. Hassan. A network partitioning methodology for distributed traffic management applications. *Transportmetrica A: Transport Science*, 10(6):518–532, 2014.

European Commission. The EU explained: Transport. Connecting Europe's citizens and businesses. Technical Report ISBN 978-92-79-42777-0, European Union, 2014.

C. Fisk. Game theory and transportation systems modelling. *Transportation Research Part B: Methodological*, 18(4):301–313, 1984.

G. Flötteröd and J. Rohde. Operational macroscopic modeling of complex urban road intersections. *Transportation Research Part B: Methodological*, 45(6):903–922, 2011.

J. R. D. Frejo and E. F. Camacho. Global versus local MPC algorithms in freeway traffic control with ramp metering and variable speed limits. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1556–1565, 2012.

C. E. Garcia, D. M. Prett, and M. Morari. Model predictive control: theory and practicea survey. *Automatica*, 25(3):335–348, 1989.

N. H. Gartner, S. B. Gershwin, J. D. Little, and P. Ross. Pilot study of computer-based urban traffic management. *Transportation Research Part B: Methodological*, 14(1): 203–217, 1980.

V. V. Gayah, X. S. Gao, and A. S. Nagle. On the impacts of locally adaptive signal control on urban network stability and the macroscopic fundamental diagram. *Transportation Research Part B: Methodological*, 70:255–268, 2014.

N. Geroliminis and C. F. Daganzo. Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. *Transportation Research Part B: Methodological*, 42(9):759–770, 2008.

N. Geroliminis, J. Haddad, and M. Ramezani. Optimal perimeter control for two urban regions with macroscopic fundamental diagrams: A model predictive approach. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):348–358, 2013.

G. Gomes and R. Horowitz. Optimal freeway ramp metering using the asymmetric cell transmission model. *Transportation Research Part C: Emerging Technologies*, 14(4):244–262, 2006.

J. Gregoire, E. Frazzoli, A. de La Fortelle, and T. Wongpiromsarn. Capacity-aware back-pressure traffic signal control. *IEEE Transactions on Control of Network Systems*, 2(2):164–173, 2014.

E. Grumert, A. Tapani, and X. Ma. Effects of a cooperative variable speed limit system on traffic performance and exhaust emissions. In *Proceedings of the 92nd Annual Meeting of the Transportation Research Board*, Washington D.C., USA, 2013.

J. Haddad, M. Ramezani, and N. Geroliminis. Model predictive perimeter control for urban areas with macroscopic fundamental diagrams. In *Proceedings of the American Control Conference*, pages 5757–5762, Los Alamitos, 2012.

H. Hadj-Salem, J. Blosseville, and M. Papageorgiou. Alinea: A local feedback control law for on ramp metering; a real life study. In *Proceedings of the third international conference on road traffic control*, pages 194–198, May 1-3 1990.

M. Hajiahmadi, R. Corthout, C. Tampére, B. De Schutter, and J. Hellendoorn. Variable speed limit control based on extended link transmission model. *Transportation Research Record: Journal of the Transportation Research Board*, 2390(1):11–19, 2013a.

M. Hajiahmadi, V. Knoop, B. De Schutter, and J. Hellendoorn. Optimal dynamic route guidance: A model predictive approach using the macroscopic fundamental diagram. In *Proceedings of the 16th IEEE Conference on Intelligent Transportation Systems*, pages 1022–1028, The Hague, The Netherlands, Oct. 6-9 2013b.

M. Hajiahmadi, J. Haddad, B. De Schutter, and N. Geroliminis. Optimal hybrid perimeter and switching plans control for urban traffic networks. *IEEE Transactions on Control Systems Technology*, 23(2):464–478, 2015a.

M. Hajiahmadi, G. S. van de Weg, C. Tampére, R. Corthout, A. Hegyi, B. De Schutter, and J. Hellendoorn. Integrated predictive control of freeway networks using the extended link transmission model. *IEEE Transactions on Intelligent Transportation Systems*, Pp(99):1–14, 2015b.

F. L. Hall and K. Agyemang-Duah. Freeway capacity drop and the definition of capacity. *Transportation Research Record: Journal of the Transportation Research Board*, (1320), 1991.

K. L. Head, P. B. Mirchandani, and D. Sheppard. Hierarchical framework for real-time traffic control. *Transportation Research Record: Journal of the Transportation Research Board*, pages 82–82, 1992.

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

A. Hegyi, B. De Schutter, and J. Hellendoorn. Model predictive control for optimal coordination of ramp metering and variable speed limits. *Transportation Research Part C: Emerging Technologies*, 13(3):185–209, 2005b.

A. Hegyi, S. P. Hoogendoorn, M. Schreuder, and Stoelhorst. The expected effectivity of the dynamic speed limit algorithm SPECIALIST - a field data evaluation method. In *Proceedings of the 13th international IEEE Conference on Intelligent Transportation Systems*, pages 1770–1775, Budapest, Hungary, Aug. 26–26 2009.

A. Hegyi, S. P. Hoogendoorn, M. Schreuder, and Stoelhorst. Dynamic speed limit control to resolve shock waves on freeways - field test results of the SPECIALIST algorithm. In *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems*, pages 519–524, Madeira, portugal, Sept. 19-22 2010.

A. Heygyi, B. Netten, M. Wang, W. Schakel, T. Schreiter, Y. Yuan, B. Van Arem, and T. Alkim. A cooperative system based variable speed limit control algorithm against jam waves an extension of the specialist algorithm. In *Proceedings of the 16th IEEE Conference on Intelligent Transportation Systems*, pages 973–978, The Hague, The Netherlands, Oct. 6-9 2013 2013.

M. Hisai and T. Usami. Optimal division of signal-coordinated arterial street into subareas. *Memoirs of the Faculty of Engineering*, 56(2):69–81, 2006.

R. Hoornman and V. Bronkhorst. *Handboek verkeerslichtenregelingen*. CROW, 2014. ISBN 978 90 6628 643 6.

P. Hunt, D. Robertson, R. Bretherton, and M. Royle. The SCOOT on-line traffic signal optimisation technique. *Traffic Engineering & Control*, 23(4):190 – 199, 1982.

A. Jamshidnejad, I. Papamichail, M. Papageorgiou, and B. De Schutter. A model-predictive urban traffic control approach with a modified flow model and endpoint penalties. *IFAC-PapersOnLine*, 49(3):147–152, 2016.

D. Jansen. Linear robust model predictive control for urban traffic networks. Technical report, Delft University of Technology, 2016. MSc thesis.

Y. Ji and N. Geroliminis. On the spatial partitioning of urban transportation networks. *Transportation Research Part B: Methodological*, 46(10):1639–1656, 2012.

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

B. S. Kerner and H. Rehborn. Experimental features and characteristics of traffic jams. *Physical Review E*, 53(2):R1297–R1300, 1996.

M. Keyvan-Ekbatani, A. Kouvelas, I. Papamichail, and M. Papageorgiou. Exploiting the fundamental diagram of urban networks for feedback-based gating. *Transportation Research Part B: Methodological*, 46(10):1393–1403, 2012.

M. Keyvan-Ekbatani, M. Papageorgiou, and V. Knoop. Controller design for gating traffic control in presence of time-delay in urban road networks. *Transportation Research Part C: Emerging Technologies*, 59:308–322, 2015a.

M. Keyvan-Ekbatani, M. Yildirimoglu, N. Geroliminis, and M. Papageorgiou. Multiple concentric gating traffic control in large-scale urban networks. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):2141 – 2154, 2015b.

A. Kotsialos and M. Papageorgiou. Efficiency and equity properties of freeway network-wide ramp metering with AMOC. *Transportation Research Part C: Emerging Technologies*, 12(6):401–420, 2004.

A. Kotsialos, M. Papageorgiou, C. Diakaki, Y. Pavlis, and F. Middelham. Traffic flow modeling of large-scale motorway networks using the macroscopic modeling tool METANET. *IEEE Transactions on Intelligent Transportation Systems*, 3(4):282–292, 2002a.

A. Kotsialos, M. Papageorgiou, M. Mangeas, and H. Hadj-Salem. Coordinated and integrated control of motorway networks via non-linear optimal control. *Transportation Research Part C: Emerging Technologies*, 10(1):65–84, 2002b.

A. Kotsialos, M. Papageorgiou, and F. Middelham. Local and optimal coordinated ramp metering for freeway networks. *Journal of Intelligent Transportation Systems*, 2005.

W. Kraus Jr, F. A. De Souza, R. C. Carlson, M. Papageorgiou, L. Dantas, E. Camponogara, E. Kosmatopoulos, and K. Aboudolas. Cost effective real-time traffic signal control using the TUC strategy. *IEEE Intelligent Transportation Systems Magazine*, 2(4):6 – 17, 2010.

R. D. Kühne. Freeway control using a dynamic traffic flow model and vehicle reidentification techniques. *Transportation Research Record: Journal of the Transportation Research Board*, (1320):251–259, 1991.

S. Lämmer and D. Helbing. Self-control of traffic lights and vehicle flows in urban road networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008 (04):P04019, 2008.

T. Le, H. Vu, Y. Nazarathy, Q. Vo, and S. P. Hoogendoorn. Linear-quadratic model predictive control for urban traffic networks. *Transportation Research Part C: Emerging Technologies*, 36:498 – 512, 2013.

T. Le, P. Kovács, N. Walton, H. Vu, L. H. Andrew, and S. P. Hoogendoorn. Decentralized signal control for urban road networks. *Transportation Research Part C: Emerging Technologies*, 58:431–450, 2015.

L. Leclercq, V. L. Knoop, F. Marczak, and S. P. Hoogendoorn. Capacity drops at merges: New analytical investigations. *Transportation Research Part C: Emerging Technologies*, 62:171–181, 2016.

P. Li, P. Mirchandani, and X. Zhou. Solving simultaneous route guidance and traffic signal optimization problem using space-phase-time hypernetwork. *Transportation Research Part B: Methodological*, 81:103–130, 2015.

M. J. Lighthill and G. B. Whitham. On kinematic waves, II. A theory of traffic flow on long crowded roads. In *Proceedings of the Royal Society of London Series a-Mathematical and Physical Sciences*, volume 229A, pages 317–345, May 1955 1955.

S. Lin, B. De Schutter, X. Yugeng, and H. Hellendoorn. Fast model predictive control for urban road networks via MILP. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):846–856, 2011.

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

J. Little. The synchronization of traffic signals by mixed-integer linear programming. *Operations Research*, 14(4):568–594, 1966.

J. D. C. Little, M. D. Kelson, and N. H. Gartner. A versatile program for setting signals on arteries and triangular networks. Technical report, Massachusetts Institute of Technology, Cambridge, 1981.

H. Lo. A novel traffic signal control formulation. *Transportation Research Part A: Policy and Practice*, 33(6):433–448, 1999.

J. Long, Z. Gao, and W. Szeto. Discretised link travel time models based on cumulative flows: formulations and properties. *Transportation Research Part B: Methodological*, 45(1):232–254, 2011.

X.-Y. Lu, P. Varaiya, R. Horowitz, D. Su, and S. E. Shladover. Novel freeway traffic control with variable speed limit and coordinated ramp metering. *Transportation Research Record: Journal of the Transportation Research Board*, (2229):55–65, 2011.

J. Luk. Two traffic-responsive area traffic control methods: SCAT and SCOOT. *Traffic engineering & control*, 25(1):14 – 22, 1984.

J. Luk, A. Sims, and P. Lowrie. SCATS-application and field comparison with a transyt optimised fixed time system. In *Proceedings of the International Conference on Road Traffic Signalling*, London, United Kingdom, 1982.

Y.-Y. Ma, Y.-C. Chiu, and X.-G. Yang. Urban traffic signal control network automatic partitioning using laplacian eigenvectors. pages 1–5, 2009.

N. Mahajan, A. Hegyi, G. S. Van de Weg, and S. P. Hoogendoorn. Integrated variable speed limit and ramp metering control against jam waves a COSCAL v2 based approach. In *Proceedings of the 18th International Conference on Intelligent Transportation Systems*, pages 1156 – 1162, Las Palmas, Spain, 2015.

P. Marcotte and J.-P. Dussault. A sequential linear programming algorithm for solving monotone variational inequalities. *SIAM Journal on Control and Optimization*, 27 (6):1260–1278, 1989.

D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

F. Middelham and H. Taale. Ramp metering in the Netherlands: an overview. In *11th IFAC symposium on transportation systems*, Delft, the Netherlands, 2006.

R. Nishi, A. Tomoeda, K. Shimura, and K. Nishinari. Theory of jam-absorption driving. *Transportation Research Part B: Methodological*, 50:116–129, 2013.

M. Papageorgiou, J. M. Blosseville, and H. Hadj-Salem. Modeling and real-time control of traffic flow on the southern part of boulevard-peripherique in Paris part 2: Coordinated on-ramp metering. *Transportation Research Part A: Policy and Practice*, 24(5):361–370, 1988.

M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12):2043–2067, 2003.

I. Papamichail and M. Papageorgiou. Traffic-responsive linked ramp-metering control. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):111–121, 2008.

M. Papgeorgiou and A. Messmer. Dynamic network traffic assignment and route guidance via feedback regulation. *Transportation Research Record: Journal of the Transportation Research Board*, 1306:49–58, 1991.

A. C. Pigou. *The economics of welfare*. Palgrave Macmillan, 2013. ISBN 1137375639.

J. B. Rawlings and D. Q. Mayne. *Model predictive control: Theory and design*. Nob Hill Pub., Madison, Wisconsin, 2009. ISBN 978-0-9759377-0-9.

R. Scarinci, A. Hegyi, and B. G. Heydecker. Analysis of traffic performance of a ramp metering strategy using cooperative vehicles. In *Proceedings of the 16th IEEE Conference on Intelligent Transportation Systems*, pages 324–329, The Hague, The Netherlands, Oct. 6-9 2013 2013.

I. Schelling, A. Hegyi, and S. P. Hoogendoorn. SPECIALIST-RM - integrated variable speed limit control and ramp metering based on shock wave theory. In *Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems*, pages 2154–2159, New York, USA, Oct. 5-7, 2010 2011.

S. E. Shladover. Effects of cooperative adaptive cruise control on traffic flow: testing drivers choices of following distances. Technical report, California PATH program, Institute of Transportation Studies University of California, Berkely, 2009.

E. Smaragdis, M. Papageorgiou, and E. Kosmatopoulos. A flow-maximizing adaptive local ramp metering strategy. *Transportation Research Part B: Methodological*, 38 (3):251–270, 2004.

M. Smith. Traffic signals in assignment. *Transportation Research Part B: Methodological*, 19(2):155–160, 1985.

E.-S. Smits, M. C. Bliemer, A. J. Pel, and B. van Arem. A family of macroscopic node models. *Transportation Research Part B: Methodological*, 74:20–39, 2015.

S. Smulders. Control of freeway traffic flow by variable speed signs. *Transportation Research Part B: Methodological*, 24(2):111–132, 1990.

F. Soriguera, I. Martínez, M. Sala, and M. Menéndez. Effects of low speed limits on freeway traffic flow. *Transportation Research Part C: Emerging Technologies*, 77: 257–274, 2017.

H. Taale and S. P. Hoogendoorn. A framework for real-time integrated and anticipatory traffic management. In *Proceedings of the 16th IEEE Conference on Intelligent Transportation Systems*, pages 449–454, The Hague, The Netherlands, Oct. 6-9 2013 2013.

H. Taale and H. Schuurman. Effecten van benutting in nederland. Technical report, TrafficQuest, 2015.

H. Taale and H. J. van Zuylen. The combined traffic assignment and control problem: An overview of 25 years of research. In *Selected Proceedings of the 9th World Conference on Transport Research*, 2001.

C. Tampère, R. Corthout, D. Cattrysse, and L. Immers. A generic class of first order node models for dynamic macroscopic simulation of traffic flows. *Transportation Research Part B: Methodological*, 45(1):289–309, 2011.

T. Tettamanti and I. Varga. Distributed traffic control system based on model predictive control. *Civil Engineering*, 54(1):3–9, 2010.

T. Tettamanti, T. Luspay, B. Kulcsár, T. Péni, and I. Varga. Robust control for urban road traffic networks. *IEEE Transactions on Intelligent Transportation Systems*, 15 (1):385–398, 2014.

S. V. Ukkusuri, G. Ramadurai, and G. Patil. A robust transportation signal control problem accounting for traffic dynamics. *Computers & Operations Research*, 37(5): 869–879, 2010.

B. Van Arem, C. J. Van Driel, and R. Visser. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):429–436, 2006.

G. S. van de Weg, A. Hegyi, J. Hellendoorn, and S. E. Shladover. Cooperative systems based control for integrating ramp metering and variable speed limits. In *Proceedings of the 93rd Annual Meeting of the Transportation Research Board*, 2014a.

G. S. van de Weg, A. Hegyi, and S. P. Hoogendoorn. Ex-ante data analysis approach for assessing the effect of variable speed limits. In *Proceedings of the 17th International Conference on Intelligent Transportation Systems*, pages 1317–1322, Qindao, China, 2014b.

G. S. van de Weg, A. Hegyi, S. P. Hoogendoorn, and B. De Schutter. Efficient model predictive control for variable speed limits by optimizing parameterized control schemes. In *Proceedings of the 18th International Conference on Intelligent Transportation Systems*, pages 1137–1142, Las Palmas, Spain, 2015.

G. S. van de Weg, M. Keyvan-Ekbatani, A. Hegyi, and S. P. Hoogendoorn. Urban network throughput optimization via model predictive control using the link transmission model. In *Proceedings of the 95th annual meeting of the Transportation Research Board*, Washington D.C., USA, 2016.

M. Van den Berg, A. Hegyi, B. De Schutter, and J. Hellendoorn. Integrated traffic control for mixed urban and freeway networks: A model predictive control approach. *European Journal of Transport and Infrastructure Research*, 7(3):223–250, 2007.

E. Van den Hoogen and S. Smulders. Control by variable-speed signs - results of the Dutch experiment. In *Proceedings of the 7th International Conference on Road Traffic Monitoring and Control*, pages 145–149, London, England, Apr 26-28 1994.

J. van der Werf, S. E. Shladover, M. A. Miller, and N. Kourjanskaia. Effects of adaptive cruise control systems on highway traffic flow capacity. *Transportation Research Record: Journal of the Transportation Research Board*, 1800:78 – 84, 2002.

P. Varaiya. Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies*, 36:177 – 195, 2013.

W. S. Vickrey. Congestion theory and transport investment. *The American Economic Review*, 59(2):251–260, 1969.

M. Wang, W. Daamen, S. P. Hoogendoorn, and B. van Arem. Potential impacts of ecological adaptive cruise control systems on traffic and environment. *IET Intelligent Transport Systems*, 8(2):77–86, 2014.

M. Wang, W. Daamen, S. P. Hoogendoorn, and B. Van Arem. Connected variable speed limits control and vehicle acceleration control to resolve moving jams. In *Proceedings of the 94th Annual Meeting of the Transportation Research Board*, Washington D.C., USA, 2015.

J. G. Wardrop and J. Whitehead. Correspondence. some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers*, 1(5):767–768, 1952.

F. Xiao, Z. S. Qian, and H. M. Zhang. Managing bottleneck congestion with tradable credits. *Transportation Research Part B: Methodological*, 56:1–14, 2013.

I. Yperman. *The link transmission model for dynamic network loading*. PhD thesis, KU Leuven, 2007.

S. K. Zegeye, B. De Schutter, J. Hellendoorn, E. A. Breunesse, and A. Hegyi. A predictive traffic controller for sustainable mobility using parameterized control policies. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1420–1429, 2012.

Y. Zhang and P. A. Ioannou. Combined variable speed limit and lane change control for highway traffic. *IEEE Transactions on Intelligent Transportation Systems*, 2016.

Z. Zhou, S. Lin, Y. Xi, D. Li, and J. Zhang. A hierarchical urban network control with integration of demand balance and traffic signal coordination. *IFAC-PapersOnLine*, 49(3):31–36, 2016.

Summary

Traffic control algorithms are not always able to efficiently utilize the network capacity causing economical and societal costs. The main complicating factor of network-wide traffic control is (simply) the size of the network, especially when controlling the traffic in an entire urban region – i.e. a densely populated area housing several millions of people. Controlling the traffic in such a region requires the coordination of several hundreds of actuators, such as variable speed limits (VSLs), ramp metering (RM), traffic lights, and route guidance. This is a challenging problem from a computational point of view due to the large amount of decision variables, but also from a theoretical point of view due to the many problem characteristics that need to be accounted for.

A promising approach to control the traffic in very large networks is to divide the network into sub-networks. A *sub-network* is defined in this dissertation as a *medium-to-large scale network* consisting of tens of kilometers of freeway or tens of intersections. The sub-network controllers are used to optimize the performance in the sub-networks while a higher level controller optimizes the flows that are exchanged between the sub-networks leading to network-wide performance improvement. In this way, the sub-network controllers can consider more detail while the higher level controller can consider more simplified or aggregated dynamics. This dissertation focuses on the design of algorithms for sub-networks in the light of a multi-level or hierarchical system as discussed above. Two types of sub-networks are considered, namely, freeway and urban sub-networks.

Ideally, a freeway or urban traffic control algorithm is able to automatically select the control signals that maximize the sub-network throughput in different (traffic) situations. Although various optimization-based algorithms have been proposed to achieve that goal, this type of algorithm has not been implemented in practice due to several reasons, namely; 1) the computational complexity of the optimization problem, 2) the noise and uncertainties involved when estimating traffic states and predicting disturbances, and 3) the not very insightful optimized control actions. In contrast to that, mainly non-optimizing control algorithms of the feedback or the feed-forward type are implemented in practice. Advantages of these algorithms are that they require little computation time, that they do not rely on demand predictions, and that they exploit simple or insightful algorithmic formulations. However, they may not be able to optimize the performance in all traffic situations.

Recent technological innovations and scientific insights provide opportunities for im-

proving both freeway and urban traffic control algorithms. Technological innovations, such as the proliferation of in-vehicle technology enabling cooperative systems, can be used to provide better detection and actuation possibilities that may be used to improve the controller performance. Similarly, scientific insights may be used to develop new algorithms that make more efficient use of existing detection and actuation possibilities. In some cases, a combined approach may be followed in which new algorithms are developed that make efficient use of new detection and actuation possibilities.

Given the network-wide traffic control problem and the opportunities to improve traffic control algorithms as discussed above, the main aim of this dissertation is **the design of computationally efficient traffic control algorithms for throughput improvement of medium-to-large scale freeway or urban traffic networks that**:

- coordinate the control actions of (different types of) actuators at different locations in the network,
- take the impact of the control actions on the network-wide performance over a time horizon into account.

The main research objective is achieved by developing several algorithms for the control of traffic in freeway networks (part I) and urban traffic networks (part II) as discussed below.

Part I – Freeway traffic control
Cooperative systems can be used to develop more efficient freeway traffic control algorithms when compared to existing, purely infrastructure-based systems. The reason for this is that using in-vehicle technology may provide more accurate and faster detection and actuation possibilities. However, not many approaches for the coordinated control of individual vehicles to control the traffic flows on an entire freeway stretch exist.

To this end, **Chapter 2** proposes a *cooperative speed control algorithm to resolve jam waves* in order to improve the freeway throughput. The algorithm – called COSCAL v1 – uses the individual vehicles as detectors and actuators assuming a 100% penetration rate. The road-side system computes based on floating car data (FCD) which driving strategy vehicles on the freeway have to follow between which locations on the freeway to resolve the jam wave and stabilize the traffic. Simulations using microscopic simulation show that the algorithm is able to improve the freeway throughput by resolving a jam wave using a negligible amount of computation time. Hence, this chapter shows that the it is possible to develop efficient algorithms for the control of traffic flows using cooperative systems.

The application of control strategies that optimize the flows between different network elements – e.g. on-ramps, off-ramps, bottlenecks, and segments – has the potential to improve the freeway performance as well. One of the main issues of this type of algorithms is balancing the required computation time and performance of the control strategy.

Hence, **Chapter 3** proposes a *computationally efficient model-based predictive control (MPC) strategy for coordinating VSLs and RM installations in order to improve the freeway throughput.* The balance between computation time and performance is improved by reducing the number of optimization variables through parameterization of the VSL and RM signal. The parameterized VSL signal consists of the speed with which the downstream and upstream boundaries of a speed-limited area propagate. The parameterized RM signal consists of the density set-points of a feedback RM strategy based on the ALINEA algorithm and the time when the settings of the feedback strategy are changed. The approach is evaluated using macroscopic simulation for two different cases, namely, when resolving a jam wave, and when preventing congestion caused by a high on-ramp demand. It is shown that the proposed MPC approach can realize throughput improvements of 12% and 10% respectively while realizing a better balance between computation time and throughput compared to a non-parameterized MPC strategy.

Part II – Urban traffic control

Improving the throughput of urban traffic networks is a complex problem due to, among others, the discontinuous nature of the intersection flows, the large number of actuators, and the characteristics of the urban traffic dynamics. To the best knowledge of the author, a computationally efficient optimization algorithm for the coordination of intersection flows that can realize good performance in all traffic regimes is currently lacking.

Therefore, **Chapter 4** proposes an *efficient linear MPC strategy for optimizing the traffic flows in order to improve the urban road network throughput*. The proposed MPC strategy uses the link transmission model (LTM) as the prediction model and aggregates the traffic flow dynamics to tens of seconds. So, instead of green-times, the fractions of green-time used by every stream are the optimization variables, which are real-valued. It is shown using macroscopic simulation that the use of the LTM leads to a better balance between computation time and realized throughput when compared to a linear MPC strategy based on the cell transmission model. It is also shown that the inclusion of upstream propagating waves leads to better throughput when compared to a linear MPC strategy based on the store-and-forward model but also that the MPC strategy requires more computation time.

The application of cooperative systems may lead to improved performance of urban traffic control algorithms. However, it may also cause an interaction effect between the chosen intersection control strategy, and the route choice of the road-users. Hence, in order to maximize the network performance, a control strategy has to account for the impact of the control signals onto the route choice and potentially control the route choice itself. However, jointly optimizing the signal timings and route choice is a computational complex problem.

Chapter 5 proposes an *efficient optimization strategy for the control of flows and routing decisions in order to improve the network throughput*. The inclusion of routing

decisions results in a non-linear prediction model and optimization problem. Therefore, an efficient optimization algorithm of the sequential linear programming (SLP) type is used and an analytic procedure to approximate the gradient in an operating point is proposed. It is shown using macroscopic simulations that the algorithm can realize a better balance between computation time and throughput when compared to applying a conventional numerical optimization algorithm.

The algorithms proposed in Chapter 4 and Chapter 5 both assume that the traffic flows at intersections are continuous. However, intersection flows are discontinuous so that directly optimizing the signal timings leads to a discontinuous optimization problem. Solving such a problem is not feasible in real-time when applied to medium-to-large scale networks. Hence, an alternative approach may be needed that can coordinate the signal timings in a network without directly optimizing the signal timings.

To this end, **Chapter 6** proposes a *hierarchical control framework to coordinate the signal timings* in order to improve the urban network throughput. The framework consists of two layers. The top layer uses the MPC strategy proposed in Chapter 4 to optimize the aggregated flows at intersections. The bottom layer consists of the individual intersection controllers which actuate at every time-step the stage that leads to the best tracking of the optimized outflows. Evaluations using macroscopic simulation are carried out to study the added value of the network coordination layer, and the impact of the timing onto the controller performance. Evaluations using microscopic simulation demonstrate the controllers ability to improve the throughput by distributing the queues over the network when compared to maximizing the outflow of the individual intersections without coordination even when subject to a larger mismatch between prediction and process model.

In **conclusion** this dissertation proposed several computationally efficient network-wide traffic control algorithms for throughput improvement of medium-to-large scale freeway or urban traffic networks. These algorithms are designed to coordinate the control actions of (different types of) actuators at different locations in the network and to take the impact of the control actions on the network-wide performance over a time-horizon into account. This is realized by exploiting new features of in-vehicle technology enabling cooperative systems to provide better detection and actuation possibilities and by using recent scientific insights to develop more efficient algorithms.

Various directions for **further research** are proposed. *First,* additional research is required to integrate the proposed algorithms into a hierarchical of multi-layer framework for the coordinated control of entire urban regions. *Second,* the algorithms proposed in this dissertation may be further improved, for instance, by further improving the balance between computation time and performance, by further exploiting the potential of in-vehicle technologies, or by studying the impact of relaxing the assumptions used in this dissertation. *Third,* recommendations for the application of concepts in practice are presented.

Samenvatting

Verkeersregelalgoritmes zijn niet altijd in staat om de capaciteit van een verkeersnetwerk volledig te benutten, wat economische en maatschappelijke kosten tot gevolg heeft. Een van de belangrijkste oorzaken hiervan is (simpelweg) de netwerk grootte. In het bijzonder als het verkeer in een gehele stedelijke regio – d.w.z. een dichtbevolkt gebied waarin miljoenen mensen wonen – geregeld dient te worden. Dit vereist namelijk de coördinatie van honderden actuatoren, zoals variabele snelheidslimieten (VSLs), toeritdoseringsinstallaties (TDIs), verkeerslichten en dynamische route informatie panelen. Dit is een uitdagend probleem vanuit een rekenkundig oogpunt vanwege het grote aantal beslisvariabelen, maar ook vanuit een theoretisch oogpunt vanwege de vele probleem-karakteristieken waarmee rekening gehouden dient te worden.

Een veelbelovende aanpak om verkeer in zeer grote netwerken te regelen is het opdelen van het netwerk in deel-netwerken. In dit proefschrift is een *deel-netwerk* gedefinieerd als een *middel- tot grootschalig netwerk* bestaande uit tientallen kilometers snelweg of tientallen kruispunten. De deel-netwerk-regelingen optimaliseren de prestatie van de deel-netwerken, terwijl een regeling op een netwerk-breed niveau de verkeersstromen optimaliseert die tussen deel-netwerken worden uitgewisseld zodat de netwerk-brede prestatie verbetert. Deze opzet zorgt ervoor dat de deel-netwerk-regelingen meer details van het verkeersproces kunnen beschouwen terwijl de netwerk-brede regeling versimpelde of geaggregeerde verkeersdynamiek kan beschouwen. Dit proefschrift richt zich op het ontwerpen van algoritmes voor deel-netwerken in het licht van een multi-level of hiërarchische regeling zoals hierboven beschreven. Twee typen deel-netwerken worden beschouwd, namelijk snelweg en stad deel-netwerken.

De ideale snelweg- of stadsregeling kiest automatisch die regelsignalen die de prestatie van het deel-netwerk optimaliseren in verschillende (verkeers-)toestanden. Alhoewel er in de literatuur verscheidene optimalisatie-gebaseerde algoritmes zijn beschreven die dit als doel hebben, is dit type algoritme nog niet in de praktijk toegepast. Dit heeft verschillende redenen, namelijk; 1) de complexiteit van het optimalisatieprobleem, 2) de meetruis en onzekerheden die de kwaliteit van schattingen van de verkeerstoestand en voorspellingen van verstoringen beïnvloeden en 3) het niet erg inzichtelijk gedrag van de geoptimaliseerde regelacties. In tegenstelling tot deze algoritmes zijn dan ook voornamelijk niet optimaliserende feedback of feed-forward algoritmes geïmplementeerd in de praktijk. De voordelen van deze algoritmes zijn dat ze maar een beperkte rekentijd nodig hebben, dat ze niet vertrouwen op een voorspelling van

de verkeersvraag en dat ze een ‘simpel’ of inzichtelijk algoritme benutten. Ze zijn echter niet altijd in staat om de prestatie in alle verkeerstoestanden te optimaliseren.

Recente technologische innovaties en wetenschappelijke inzichten bieden kansen om snelweg- en stadsverkeersregelingen te verbeteren. Technologische innovaties zoals in-voertuig technologie die coöperatieve systemen mogelijk maken, kunnen worden benut om betere detectie en actuatie mogelijkheden te creëren die de prestatie van verkeersregelalgoritmes kunnen verbeteren. Wetenschappelijke inzichten kunnen worden gebruikt om nieuw algoritmes te ontwikkelen die efficiënter gebruik maken van bestaande detectie en actuatie mogelijkheden. Daarnaast is in sommige gevallen een gecombineerde aanpak mogelijk waarin nieuwe algoritmes worden ontwikkeld die efficiënt gebruik maken van nieuwe detectie en actuatie mogelijkheden.

Gegeven het netwerk-brede verkeersregelprobleem en de kansen om verkeersregelalgoritmes te verbeteren zoals hierboven besproken is het doel van dit proefschrift om **rekenkundig efficiënte regelalgoritmes te ontwerpen die de doorstroming van middel- tot grootschalige snelweg- of stadsverkeersnetwerken bevorderen welke**:

- de regelacties van (verschillende types) actuatoren op verschillende plekken in het netwerk coördineren,
- de invloed van de regelacties op de netwerk-brede prestatie over een tijdshorizon in acht nemen.

Hiertoe worden verschillende algoritmes om het verkeer in snelwegnetwerken (deel I) en stadsnetwerken (deel II) te regelen ontwikkeld, zoals hieronder besproken.

Deel I – Snelwegverkeersregelingen
Coöperatieve systemen kunnen benut worden om snelwegverkeersregelingen te ontwikkelen die efficiënter zijn dan systemen die volledig gebaseerd zijn op wegkanttechnologie. De reden hiervoor is dat het gebruik van coöperatieve systemen nauwkeurigere en snellere detectie en actuatie mogelijk maakt. Er bestaan echter nog niet veel algoritmes die verkeersstroom op een stuk snelweg regelen door middel van het coördineren van het gedrag van individuele voertuigen.

Hoofdstuk 2 presenteert daarom een *coöperatief snelheidsregel-algoritme om filegolven op te lossen* zodat de doorstroming van de snelweg als geheel verbetert. Dit algoritme – genaamd COSCAL v1 – gebruikt de individuele voertuigen als detectoren en actuatoren uitgaande van een penetratie graad van 100%. Het wegkantsysteem berekent aan de hand van floating car data (FCD) welke rij-strategieën voertuigen moeten volgen tussen welke locaties om de file op te lossen en de verkeersstroom te stabiliseren. Evaluaties uitgevoerd met microscopische simulatie laten zien dat het algoritme in staat is om de doorstroming te verbeteren door een filegolf op te lossen gebruikmakend van een minieme hoeveelheid rekentijd. Dit hoofdstuk toont dan ook aan de het mogelijk is om efficiënte algoritmes voor coöperatieve systemen te ontwikkelen om de verkeersafwikkeling op de snelweg te verbeteren.

Het optimaliseren van de verkeersstromen tussen verschillende netwerk elementen – zoals op- en afritten, knelpunten, en stukken snelweg – heeft daarnaast ook potentie om de doorstroming te verbeteren. Een van de belangrijkste problemen van dit type algoritmes is het vinden van een goede balans tussen benodigde rekentijd en prestatie.

Derhalve stelt **Hoofdstuk 3** een *rekenkundig efficiënte, model-gebaseerd voorspellende regeling (MPC) voor die de doorstroming verbetert door de regelsignalen van VSLs en TDIs te coördineren.* De balans tussen rekentijd en doorstroming is verbeterd door het aantal beslisvariabelen te verminderen met behulp van parameterisatie van het VSL en TDI signaal. De aanpak is geëvalueerd met behulp van macroscopische simulatie voor twee verschillende casussen, namelijk het oplossen van een filegolf en het voorkomen van file veroorzaakt door een te hoge verkeersvraag op de toerit. De aanpak is in staat doorstromingsverbeteringen van respectievelijk 12% en 10% te behalen en realiseert een betere balans tussen rekentijd en doorstroming vergeleken met een niet-geparameteriseerde MPC strategie.

Deel II – Stadsverkeersregelingen
Het verbeteren van de doorstroming van een stadsverkeersnetwerk is een complex probleem vanwege, onder andere de grote hoeveelheid actuatoren en de complexe karakteristieken van de verkeersafwikkeling. Voor zover bekend bij de auteur van dit proefschrift bestaan er bijna geen aanpakken die met behulp van een rekenkundig efficiënt optimalisatie algoritme de verkeersstromen in een stadsverkeersnetwerk coördineren zodat er een goede doorstroming in alle verkeerstoestanden kan worden gerealiseerd.

Hoofdstuk 4 presenteert daarom *een efficiënte lineaire MPC strategie om de verkeersstromen te optimaliseren zodanig dat de doorstroming van een stadsverkeersnetwerk verbetert.* De voorgestelde aanpak gebruikt het link transmissie model (LTM) als voorspellingsmodel en aggregeert de verkeersdynamiek naar tientallen seconden. De beslisvariabelen zijn hierdoor de percentages groentijd benut door elke stroom welke reëele waarden hebben. Evaluaties met behulp van macroscopische simulaties laten zien dat het gebruik van het LTM tot een betere balans tussen rekentijd en prestatie leidt vergeleken met een lineaire MPC aanpak gebaseerd op het cell transmissie model. Daarnaast blijkt dat het meenemen van stroomopwaarts propagerende golven in het voorspel model tot een betere doorstroming maar een hogere rekentijd, vergeleken met een aanpak gebaseerd op het store-and-forward model.

Het gebruik van coöperatieve systemen kan mogelijk de prestatie van stadsverkeersregelingen verbeteren. Het kan echter ook een interactie effect veroorzaken tussen de gekozen verkeersregeling en de routekeuze van de weggebruiker. Een verkeersregeling die als doel heeft de doorstroming te maximaliseren dient dan ook rekening te houden met de invloed van de verkeersregeling op de routekeuze of moet mogelijk zelfs de routekeuze direct beïnvloeden. Het gezamenlijk optimaliseren van de routekeuze en de verkeersstromen leidt echter tot een rekenkundig complex optimalisatie probleem.

Hoofdstuk 5 presenteert daarom een *efficiënt optimalisatie algoritme om de verkeersstromen en routekeuzes zodanig te regelen dat de doorstroming van een stadsverkeer-*

netwerk verbetert. Het toevoegen van routekeuze leidt tot een niet-lineair optimalisatie probleem. Er wordt daarom gebruik gemaakt van een efficiënt optimalisatie algoritme van het sequentieel lineair programmeer (SLP) type in combinatie met een analytische procedure om de gradiënt van het optimalisatie probleem te bepalen. Evaluaties met behulp van macroscopische simulatie laten zien dat het algoritme in staat is een betere balans tussen rekentijd en doorstroming te realiseren in vergelijking tot een conventioneel numeriek optimalisatie algoritme.

De algoritmes gepresenteerd in Hoofdstuk 4 en Hoofdstuk 5 nemen beiden aan dat de verkeersstromen over kruispunten continu zijn. In de praktijk zijn deze echter discontinu zodat het direct optimaliseren van de groentijden een discontinu optimalisatieprobleem oplevert. Het oplossen van een dergelijk optimalisatieprobleem kost teveel rekentijd voor praktische toepassing. Er is dan mogelijk ook een alternatieve aanpak nodig om de kruispuntregelingen in een verkeersnetwerk te coördineren zonder expliciet de groentijden te optimaliseren.

Hoofdstuk 6 presenteert daarom *een hiërarchische regelaanpak om de groentijden te coördineren* bestaande uit twee lagen. De bovenste laag gebruikt de MPC aanpak uit Hoofdstuk 4 om de geaggregeerde verkeersstromen in het netwerk te optimaliseren. De onderste laag bestaat uit de individuele kruispuntregelingen welke op elke regel tijdstap die richtingen groen geven die ervoor zorgen dat de geoptimaliseerde verkeersstromen berekend door de bovenste laag zo goed mogelijk benaderd worden. Evaluaties met behulp van macroscopische simulatie geven inzicht in de toegevoegde waarde van de bovenste laag die zorgt voor coördinatie tussen de kruispunten en inzicht in de invloed van de sample tijd keuze op de prestatie van de regeling. Evaluaties met behulp van microscopische simulatie laten zien dat de regeling een betere doorstroming kan realiseren door de wachtrijen over het netwerk te verdelen in vergelijking tot een regeling die de uitstroom van de individuele kruispunten optimaliseert zonder coördinatie, zelfs als er een grotere fout tussen het voorspel- en procesmodel zit.

Verscheidene rekenkundig efficiënte algoritmes voor het verbeteren van de doorstroming van middel- tot grootschalige verkeersnetwerken zijn ontwikkeld in dit proefschrift. Deze algoritmes zijn ontworpen om de regelacties van (verschillende types) actuatoren op verschillende plekken in het netwerk te coördineren en de invloed van de regelacties op de netwerk-brede prestatie over een tijdshorizon in acht nemen. Dit is gerealiseerd door gebruik te maken van recente wetenschappelijke inzichten en de nieuwe detectie- en actuatie-mogelijkheden die coöperatieve systemen bieden.

Dit proefschrift presenteert verschillende richtingen voor **vervolg onderzoek**. *Ten eerste* is aanvullend onderzoek nodig om de gepresenteerde algoritmes te integreren in een hiërarchisch of multi-level raamwerk voor het gecoördineerd regelen van stedelijke regio's. *Ten tweede* worden er aanbevelingen gedaan om de balans tussen rekentijd en prestatie van de gepresenteerde algoritmes nog verder te verbeteren. *Tot slot* worden er aanbevelingen gedaan voor aanvullend onderzoek dat nodig is om de concepten in de praktijk toe te passen.

About the author

Goof Sterk van de Weg was born in Dordrecht, the Netherlands on January 12, 1989. He finished the Dutch pre-university education program 'Voorbereidend Wetenschappelijk Onderwijs' from 'het Thuredrecht college' in Dordrecht in 2007. In that same year he started the BSc in Mechanical Engineering at the Delft University of Technology (TU Delft) and he completed the propaedeutic exam with distinction – i.e. top 5% of the students – in 2008. During his BSc he took one year off in the academic year of 2009-2010 to be part of the board of the study association of Mechanical Engineering 'Gezelschap Leeghwater' for which the TU Delft awarded a full academic year scholarship (1 FTE). During that year his main responsibility representing the interests of about 1000 students with the faculty of Mechanical, Maritime, and Materials Engineering (3ME). He received the BSc degree with distinction in 2011 and started his MSc degree in Systems and Control at the Delft Center for Systems and Control (DCSC) department at the 3ME faculty. As part of the MSc program, he carried out a 1 year MSc thesis project that studied the application of cooperative systems to resolve a jam wave using in-vehicle speed limits and ramp metering. As part of this project he visited the PATH research institute at the University of California of Berkeley for four months. He was awarded the 'Justus en Louise van Effen' scholarship which is given to excellent students who wish to conduct research at a top-20 Engineering and Technology University abroad. He received is MSc degree in Systems and Control with distinction on the 26th of June in 2013.

He started his PhD research at the department of Transport & Planning (T&P) at the faculty of Civil Engineering and Geosciences (CEG) at the TU Delft in October 2013 with Andreas Hegyi as his daily supervisor and Serge Hoogendoorn as his promotor. His PhD research focused on the design of algorithms that improve the throughput of road traffic networks. This research is part of the research programme 'The Application of Operations Research in Urban Transport', which is (partly) financed by the

Netherlands Organisation for Scientific Research (NWO). He co-authored two research reports for the Verkeersonderneming studying the applicability of a speed control algorithm on the A13 freeway in the Netherlands. As part of the PhD research project he visited the group of Hai Le Vu at the Swinburne University of Technology in Melbourne, Australia for 3,5 months at the start of 2016.

Besides research related tasks, Goof also gave several lectures in the courses of Andreas Hegyi and Serge Hoogendoorn. He supervised 6 MSc thesis students together with Andreas Hegyi and Victor Knoop of the T&P department and Bart De Schutter of the DCSC department. He completed the doctoral education program of the Graduate School of the TU Delft. He also represented the PhD students at the general board meeting of the T&P department. He worked part-time (0.2 FTE) as an advisor at Arane Adivseurs in Gouda, the Netherlands during the period from September 2016 to March 2017 in his spare time.

List of Publications

Journal articles

1. M. Hajiahmadi, **G.S. van de Weg**, C. Tampère, R. Corthout, A. Hegyi, B. De Schutter, and H. Hellendoorn. Integrated predictive control of freeway networks using the extended link transmission model. *IEEE Transactions on Intelligent Transportation Systems*, Pp(99):114, 2015b.

The following articles are currently under review:

2. **G.S. van de Weg**, A. Hegyi, B. De Schutter, and S.P. Hoogendoorn, Efficient MPC for freeway throughput improvement by parameterization of ALINEA and a speed-limited area. *Transactions on Intelligent Transportation Systems*, submitted 2017-2-17.
3. **G.S. van de Weg**, M. Keyvan-Ekbatani, A. Hegyi, and S.P. Hoogendoorn, Linear MPC-based Urban Traffic Control using the Link Transmission Model. *Transactions on Intelligent Transportation Systems*, submitted 2017-6-12.
4. **G.S. van de Weg**, E.-S. Smits, H. Taale, A. Hegyi, B. De Schutter, and S.P. Hoogendoorn, Efficient Joint Optimization of Routing and Intersection Flows using the Link Transmission Model. *Transportation Research Part C*, submitted 2017-03-25.
5. **G.S. van de Weg**, H.L. Vu, A. Hegyi, and S.P. Hoogendoorn, A Hierarchical Control Framework for Coordination of Intersection Signal Timings in all Traffic Regimes. *Transactions on Intelligent Transportation Systems*, submitted 2017-4-13.

The following article is being prepared for submission:

6. **G.S. van de Weg**, A. Hegyi, S.E. Shladover, X.-Y. Yun, D. Chen, and S.P. Hoogendoorn, COSCAL v1: A cooperative speed control algorithm for resolving jam waves. To be submitted.

Peer reviewed conference contributions

1. **G.S. van de Weg**, A. Hegyi, H. Taale, S.P. Hoogendoorn, B. De Schutter. Efficient optimization of aggregated traffic signal control, taking route

flows into account. In *Triennial Symposium on Transportation Analysis, TRISTAN IX*, Oranjestad, Aruba, the Netherlands, 2016.

2. **G.S. van de Weg**, M. Keyvan-Ekbatani, A. Hegyi, and S. Hoogendoorn. Urban network throughput optimization via model predictive control using the link transmission model. In *Proceedings of the 95th annual meeting of the Transportation Research Board*, Washington D.C., USA, 2016.
3. **G.S. van de Weg**, A. Hegyi, S.P. Hoogendoorn, and B. De Schutter. Efficient model predictive control for variable speed limits by optimizing parameterized control schemes. In *Proceedings of the 18th International Conference on Intelligent Transportation Systems*, pages 11371142 Las Palmas, Spain, 2015.
4. N. Mahajan, A. Hegyi, **G.S. Van de Weg**, and S.P. Hoogendoorn. Integrated variable speed limit and ramp metering control against jam waves a COSCAL v2 based approach. In *Proceedings of the 17th International Conference on Intelligent Transportation Systems*, pages 1156 1162, Las Palmas, Spain, 2015.
5. **G.S. van de Weg**, A. Hegyi, and S.P. Hoogendoorn. Ex-ante data analysis approach for assessing the effect of variable speed limits. In *Proceedings of the 16th International Conference on Intelligent Transportation Systems*, pages 13171322, Qindao, China, 2014.
6. **G.S. van de Weg**, A. Hegyi, J. Hellendoorn, and S.E. Shladover. Cooperative systems based control for integrating ramp metering and variable speed limits. In *Proceedings of the 93rd Annual Meeting of the Transportation Research Board*, Washington D.C., USA 2014.

Technical reports

1. **G.S. van de Weg** and A. Hegyi. Voorspellende verkeerslichten: de volgende stap in stedelijk verkeersmanagement. *NM Magazine*, 4 (1), 2016.
2. **G.S. van de Weg** and A. Hegyi. Resolving moving jams on the A13; development and evaluation of extensions of COSCAL v2. Technical Report, TU Delft, May 2014.
3. Hegyi, A., and **G.S. Van de Weg**. Spoken bestaan niet: Hoe filegolven ontstaan en hoe je ze kunt voorkomen. *NM Magazine*, 9 (1), 2014.
4. **G.S. van de Weg**, A. Hegyi, and S. P. Hoogendoorn. "Robust, Optimal, Predictive, and Integrated Road Traffic Control: Research proposal. *Joint Chinese-Dutch Seminar on Transportation Management and Travel Behaviour for Urban Emergencies: Past, Present, and Future Research*, Shanghai, China, 23-25 June 2014.
5. **G.S. van de Weg** and A. Hegyi. Resolvability analysis of moving jams on the A13. Technical Report, TU Delft, January 2014.

TRAIL Thesis Series

The following list contains the most recent dissertations in the TRAIL Thesis Series. For a complete overview of more than 150 titles see the TRAIL website: www.rsTRAIL.nl.

The TRAIL Thesis Series is a series of the Netherlands TRAIL Research School on transport, infrastructure and logistics.

Weg, G.S. van de, *Efficient Algorithms for Network-wide Road Traffic Control*, T2017/11, October 2017, TRAIL Thesis Series, the Netherlands

He, D., *Energy Saving for Belt Conveyors by Speed Control*, T2017/10, July 2017, TRAIL Thesis Series, the Netherlands

Bešinović, N., *Integrated Capacity Assessment and Timetabling Models for Dense Railway Networks*, T2017/9, July 2017, TRAIL Thesis Series, the Netherlands

Chen, G., *Surface Wear Reduction of Bulk Solids Handling Equipment Using Bionic Design*, T2017/8, June 2017, TRAIL Thesis Series, the Netherlands

Kurapati, S., *Situation Awareness for Socio Technical Systems: A simulation gaming study in intermodal transport operations*, T2017/7, June 2017, TRAIL Thesis Series, the Netherlands

Jamshidnejad, A., *Efficient Predictive Model-Based and Fuzzy Control for Green Urban Mobility*, T2017/6, June 2017, TRAIL Thesis Series, the Netherlands

Araghi, Y., *Consumer Heterogeneity, Transport and the Environment*, T2017/5, May 2017, TRAIL Thesis Series, the Netherlands

Kasraian Moghaddam, D., *Transport Networks, Land Use and Travel Behaviour: A long term investigation*, T2017/4, May 2017, TRAIL Thesis Series, the Netherlands

Smits, E.-S., *Strategic Network Modelling for Passenger Transport Pricing*, T2017/3, May 2017, TRAIL Thesis Series, the Netherlands

Tasseron, G., *Bottom-Up Information Provision in Urban Parking: An in-depth analysis of impacts on parking dynamics*, T2017/2, March 2017, TRAIL Thesis Series, the Netherlands

Halim, R.A., *Strategic Modeling of Global Container Transport Networks: Exploring the future of port-hinterland and maritime container transport networks*, T2017/1, March 2017, TRAIL Thesis Series, the Netherlands

Olde Keizer, M.C.A., *Condition-Based Maintenance for Complex Systems: Coordinating maintenance and logistics planning for the process industries*, T2016/26, December 2016, TRAIL Thesis Series, the Netherlands

Zheng, H., *Coordination of Waterborn AGVs*, T2016/25, December 2016, TRAIL Thesis Series, the Netherlands

Yuan, K., *Capacity Drop on Freeways: Traffic dynamics, theory and Modeling*, T2016/24, December 2016, TRAIL Thesis Series, the Netherlands

Li, S., *Coordinated Planning of Inland Vessels for Large Seaports*, T2016/23, December 2016, TRAIL Thesis Series, the Netherlands

Berg, M. van den, *The Influence of Herding on Departure Choice in Case of Evacuation: Design and analysis of a serious gaming experimental set-up*, T2016/22, December 2016, TRAIL Thesis Series, the Netherlands

Luo, R., *Multi-Agent Control of urban Transportation Networks and of Hybrid Systems with Limited Information Sharing*, T2016/21, November 2016, TRAIL Thesis Series, the Netherlands

Campanella, M., *Microscopic Modelling of Walking Behavior*, T2016/20, November 2016, TRAIL Thesis Series, the Netherlands

Horst, M. van der, *Coordination in Hinterland Chains: An institutional analysis of port-related transport*, T2016/19, November 2016, TRAIL Thesis Series, the Netherlands

Beukenkamp, W., *Securing Safety: Resilience time as a hidden critical factor*, T2016/18, October 2016, TRAIL Thesis Series, the Netherlands

Mingardo, G., *Articles on Parking Policy*, T2016/17, October 2016, TRAIL Thesis Series, the Netherlands

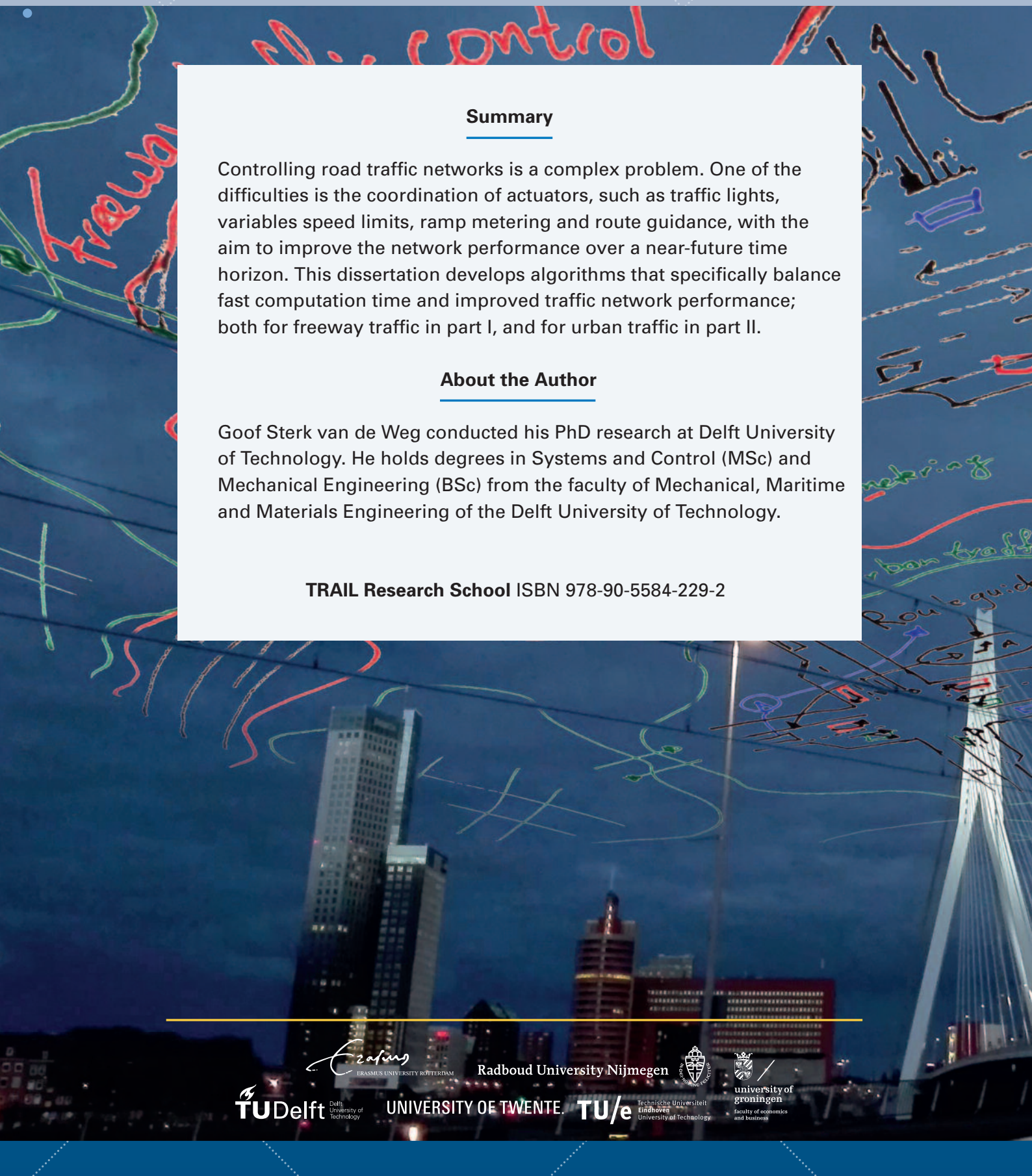
Duives, D.C., *Analysis and Modelling of Pedestrian Movement Dynamics at Large-scale Events*, T2016/16, October 2016, TRAIL Thesis Series, the Netherlands

Wan Ahmad, W.N.K., *Contextual Factors of Sustainable Supply Chain Management Practices in the Oil and Gas Industry*, T2016/15, September 2016, TRAIL Thesis Series, the Netherlands

Liu, X., *Prediction of Belt Conveyor Idler Performance*, T2016/14, September 2016, TRAIL Thesis Series, the Netherlands

Gaast, J.P. van der, *Stochastic Models for Order Picking Systems*, T2016/13, September 2016, TRAIL Thesis Series, the Netherlands

TRAIL



Summary

Controlling road traffic networks is a complex problem. One of the difficulties is the coordination of actuators, such as traffic lights, variables speed limits, ramp metering and route guidance, with the aim to improve the network performance over a near-future time horizon. This dissertation develops algorithms that specifically balance fast computation time and improved traffic network performance; both for freeway traffic in part I, and for urban traffic in part II.

About the Author

Goof Sterk van de Weg conducted his PhD research at Delft University of Technology. He holds degrees in Systems and Control (MSc) and Mechanical Engineering (BSc) from the faculty of Mechanical, Maritime and Materials Engineering of the Delft University of Technology.

TRAIL Research School ISBN 978-90-5584-229-2

Erasmus University Rotterdam · Radboud University Nijmegen · TU Delft Delft University of Technology · University of Twente. · TU/e Technische Universiteit Eindhoven University of Technology · university of groningen faculty of economics and business