Predictive Traffic Signal Control under Uncertainty
Analyzing and Reducing the Impact of Prediction Errors

Poelman, M.C.

DOI
10.4233/uuid:58d1eb12-ad5f-4f6b-8aa9-13a6d5cc096c

Publication date
2024

Document Version
Final published version

Citation (APA)
Poelman, M. C. (2024). *Predictive Traffic Signal Control under Uncertainty: Analyzing and Reducing the Impact of Prediction Errors*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:58d1eb12-ad5f-4f6b-8aa9-13a6d5cc096c

Predictive Traffic Signal Control under Uncertainty

Analyzing and Reducing the Impact of Prediction Errors

Muriel Celeste Poelman

Delft University of Technology

This doctoral thesis is part of the research project MiRRORS (project number 16270) sponsored by the Dutch Foundation for Scientific Research NWO / Applied Sciences.

Predictive Traffic Signal Control under Uncertainty

Analyzing and Reducing the Impact of Prediction Errors

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus Prof.dr.ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates,
to be defended publicly on
Wednesday 23 October 2024 at 12:30

by

Muriel Celeste POELMAN
Master of Science in Applied Mathematics
Delft University of Technology, the Netherlands
born in Arnhem, the Netherlands

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof.dr.ir. J.W.C. van Lint,	Delft University of Technology, promotor
Prof.dr.ir. A. Verbraeck,	Delft University of Technology, promotor
Dr.ir. A. Hegyi,	Delft University of Technology, copromotor

Independent members:

Prof.dr.ir. B.H.K. De Schutter,	Delft University of Technology
Prof.dr. M.M. de Weerdt,	Delft University of Technology
Prof.dr. E.I. Vlahogianni,	National Technical University of Athens
Prof.dr. F. Viti,	University of Luxembourg
Prof.dr.ir. S.P. Hoogendoorn,	Delft University of Technology, reserve member

TRAIL Thesis Series no. T2024/6, the Netherlands Research School TRAIL

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

ISBN: 978-90-5584-344-2

Printed in the Netherlands

For my husband and my children,
whose endless love keeps me going.

Acknowledgements

More than six years ago I started this PhD research. As an applied traffic engineer, I had the opportunity to do science, learning new methods, which gave me a fresh look at traffic models and systems. It has been an intensive period, with a lot of ups and downs. Now I have completed this thesis, I'm filled with joy, a little bit of proud, and a sigh of relief. This thesis would not have been possible without the contribution of others. I would like to thank all the committee members for their time and valuable comments to improve the contents of this thesis. I would like to thank my family, friends and colleagues for their support and for showing interest in my work and well-being during this intensive period. I would like to express my gratitude to a few people in special (continuing in Dutch).

Graag wil ik mijn promotoren Hans en Alexander en copromotor Andreas bedanken voor de begeleiding. Bedankt voor alle interessante inhoudelijke discussies die we samen gehad hebben over het onderzoek. Hier kijk ik met een goed gevoel op terug en ik heb veel van jullie geleerd. Hans, bedankt dat je als promotor mij de mogelijkheid hebt gegeven om, op latere leeftijd en parttime vanuit een bedrijf, dit PhD traject te doen. Bedankt voor je vertrouwen en uiteindelijke keuze voor mij in het MiRRORS project. Je aanstekelijke enthousiasme heeft mij zeker geholpen om het project tot een succesvol einde te brengen. Alexander, bedankt dat je mij wilde begeleiden als mijn andere promotor en de stabiele factor was vooral in de laatste fase van mijn PhD traject. Bedankt voor je altijd scherpe en kritische blik en positieve opbouwende commentaar. Dit gaf me altijd vertrouwen dat ik op de goede weg was om de eindstreep te halen. Andreas, bedankt dat je mijn copromotor en dagelijkse begeleider wilde zijn. Bedankt voor de persoonlijke gesprekken (over andere dingen dan het onderzoek alleen) en het in de gaten houden van mijn energie niveau. Je prettige begeleiding en structurele feedback gaf houvast om dit niet altijd makkelijke PhD traject tot een goed einde te brengen.

Ook wil ik mijn leidinggevenden door de jaren heen bij Royal HaskoningDHV bedanken voor de ruimte om dit onderzoek naast mijn andere werkzaamheden te kunnen doen. Bart, in het bijzonder, bedankt dat je mij de kans hebt gegeven om dit PhD traject aan de universiteit te volgen. Verder wil ik mijn directe collega's bedanken voor hun interesse in mijn onderzoek. Suzanne en Geertje, bedankt voor jullie deelname aan de gebruikerscommissie van het MiRRORS project en jullie betrokkenheid. Fijn dat ik bij jullie terecht kon, als het even iets minder ging. Kartik, bedankt voor de technische support bij het bouwen van mijn prototyping omgeving. M'n hardloopmaatje, Esther, bedankt voor je luisterend oor, en de gezellige kletspraatjes tijdens onze loopjes in het bos. Collega's, bedankt voor de opbeurende gesprekken bij het koffieautomaat. Jullie aandacht was zeer welkom om dit intensieve traject vol te kunnen houden.

Tot slot wil ik mijn familie bedanken voor hun steun. M’n lieve familie, bedankt dat ik bij jullie terecht kon om m’n ups en downs te delen. M’n ouders, in het bijzonder, bedankt dat jullie er altijd voor me zijn als het nodig is. M’n kinderen, Perijn en Elora, bedankt voor jullie liefde en opbeurende knuffels. Bedankt voor alle fijne momenten samen met ons gezin, de boswandelingen, ‘t knutselen, de vele spelletjes, en filmavonden, waardoor ik het werk even kon vergeten en uiteindelijk alles kon relativeren. M’n man, Jarno, bedankt dat je altijd voor me klaarstaat en ik altijd op je kan rekenen. We hebben een veeleisende periode achter de rug, allebei tegelijkertijd ons PhD traject, daarnaast de verbouwing van ons huis, de zorg voor je vader, en het thuisonderwijs tijdens COVID. Zonder jou was ik hier niet doorheen gekomen. Het is ons samen gelukt, we hebben het allebei gered, iets om trots op te zijn! En nu tijd voor een nieuw project samen, dat vakantiehuisje?

Muriel Poelman
Bilthoven, September 2024

Contents

1 Introduction ... 1
1.1 Research background on predictive traffic signal control ... 2
1.1.1 Adaptive traffic signal control ... 2
1.1.2 Prediction methods ... 3
1.1.3 Prediction uncertainties ... 4
1.1.4 Robust traffic signal control ... 6
1.2 Research objective ... 8
1.3 Research questions ... 8
1.4 Research approach ... 10
1.5 Scientific contributions ... 11
1.5.1 Prediction quality ... 11
1.5.2 Robust control ... 12
1.5.3 Heuristic control methods ... 12
1.5.4 Real-time prototyping ... 12
1.6 Thesis outline ... 13

2 Sensitivity analysis on the impact of prediction errors in local intersection control ... 15
2.1 Introduction ... 16
2.2 Problem description ... 16
2.3 State of the art ... 18
2.4 Experimental framework ... 18
2.5 Case: predictive control ... 20
2.5.1 Predictive control model ... 21
2.5.2 Experimental settings control scenario ... 24
2.5.3 Experimental results sensitivity analysis ... 26
2.5.4 Design guidelines ... 30
2.6 Conclusions and recommendations ... 32

3 Sensitivity analysis on the impact of prediction errors in network control ... 35
3.1 Introduction ... 36
3.2 Methods ... 38
3.2.1 Experimental setup: general approach sensitivity analysis ... 38
3.2.2 Technical setup: model predictive control system ... 42
3.2.3 Traffic scenario: corridor with spillback ... 51
3.3 Results ... 54
3.3.1 Reference: perfect world ... 54
3.3.2 Aggregation errors ... 56
3.3.3 Biases ... 58
3.4 Discussion ... 62
3.5 Conclusion ... 64

4 Robust approach to reduce the impact of prediction uncertainties in network control ... 67
4.1 Introduction ... 68
4.2 Background ... 70
4.2.1 Robust offline fixed-time control ... 70
4.2.2 Robust real-time adaptive control ... 73
4.3 Approach ... 75
4.3.1 Robust real-time structure-free model-based predictive control ... 75
4.3.2 Mathematical formulation robust control system ... 77
4.3.3 Simulated traffic case ... 82
4.4 Results ... 86
4.4.1 Non-robust control as a reference ... 86
4.4.2 Potential of full robust control ... 89
4.4.3 Efficiency of robust control heuristic ... 89
4.4.4 Overprotectiveness ... 91
4.5 Conclusion ... 92

5 Prototyping environment for real-time optimizing predictive traffic signal control applications ... 95
5.1 Introduction ... 96
5.2 Approach ... 98
5.2.1 Modules ... 99
5.2.2 Threads ... 102
5.2.3 Methods ... 107
5.3 Conclusion ... 115

6 Conclusion 119
6.1 Key findings 120
6.1.1 Prediction quality 120
6.1.2 Control robustness.. 122
6.2 Scientific contributions.. 123
6.2.1 Prediction quality 123
6.2.2 Robust control 124
6.2.3 Heuristic control methods 124
6.2.4 Real-time prototyping 125
6.3 Practical implications 126
6.4 Scientific recommendations 128

Bibliography 131

Summary 135

Samenvatting 137

About the author.. 139

TRAIL Thesis Series 141

Chapter 1

Introduction

Urban areas are getting increasingly crowded, and it will be a challenge to keep cities livable and accessible. Traffic management in urban areas is important to reduce congestion and improve accessibility. Traffic signal controllers are often applied as part of such a traffic management system. Traffic signal control systems are not only used to control the traffic flows safely and efficiently at a single intersection but are often used (in combination with route guidance) in coordinated traffic management to control traffic flows on a network level. To efficiently control traffic in a network, real-time data is used to react properly to the actual traffic conditions. Network control can become even more effective if knowledge of future traffic conditions can be used to pro-actively control the traffic. Pro-active traffic management, and more specifically pro-active traffic signal control, will contribute to the reduction of congestion in urban areas.

Traffic signal control systems are becoming more advanced [1], [2]. New technologies in the field of communicating vehicles and intelligent traffic lights make the traffic signal control system more interactive [3], [4], [5]. New information and data sources become available, to provide the necessary insight in the traffic conditions, varying from historical to real-time data, and from location-based data (like loop detectors) to floating-car data of individual vehicles [6], [7]. Advanced traffic state estimation [8] and prediction methods [9], [10] are developed based on these data. Some of these prediction techniques have already been proposed in traffic signal control methods to pro-actively control traffic based on predicted traffic conditions [2]. In general, traffic signal controllers are becoming more adaptive [11], reconsidering control decisions more often, using more information, and allowing more degrees of control freedom, such that the control decision matches the upcoming traffic pattern better up to the level of individual vehicle movements. These advanced highly adaptive predictive traffic signal control systems may have the potential to efficiently control traffic in a network increasing the control performance.

However, since these control systems rely on more information, highly adaptive predictive controllers may be vulnerable to information errors that occur in practice. In real life, prediction methods are not perfect and will contain errors. These prediction errors will eventually affect the control decision and the performance of the control system. To fully benefit from prediction techniques in traffic signal control systems in practice, the performance impact of prediction uncertainties needs to be considered in the control design. Insights are needed into possible performance loss, i.e., when the prediction errors become too large and affect control performance, and which type of prediction errors lead to the largest performance loss. And, vice versa, insights are needed into possible performance gain, i.e., which improvements in prediction accuracy lead to the largest performance gain, and when the predictions are accurate enough to let the controller function properly. Besides, if predictions are hard to improve in practice, predictive control systems need to be designed that protect against large performance loss due to uncertainties, so called robust control. Until now, prediction uncertainties have been barely considered in existing predictive traffic signal control methods, therefore:

In this thesis, highly adaptive predictive traffic signal control systems are studied considering prediction uncertainties. In specific, the effect of prediction errors on the control performance is analyzed and robust methods to reduce the performance impact of prediction uncertainties are designed.

In the remainder of this introduction, first the research background on predictive traffic signal control is described in more detail including the presence of uncertainties. Then the research objective, research questions and research approach are outlined, and the scientific contributions obtained in the research are formulated. The thesis outline is provided as a reader's guideline to understand the coherence between the different chapters of this thesis.

1.1 Research background on predictive traffic signal control

1.1.1 Adaptive traffic signal control

There is a wide variety of signalized traffic control types, as described in [1], [11], [12] and more recently in [2], [3], [4], [5]. Signalized traffic control methods can roughly be divided into two categories, fixed-time control, and traffic-responsive control. In fixed-time control, the control plan is optimized offline based on historical demand data. In traffic-responsive (or adaptive) control, the control plan is adapted online based on real-time traffic data. The adaptivity level of the controller has been increasing over the years [11] to better adapt the control plan to the actual traffic pattern and improve control performance in a network. The controllers' adaptivity, as considered in this thesis, consists of different aspects:

- Control update: The controller is more adaptive if the controller updates the control plan more often based on new available data. Note that fixed-time control uses a fixed control plan that is predetermined offline. Traffic-responsive control adapts the control plan online based on real-time data. The control update mostly is a closed-loop *feedback system* that monitors the effect of the control actions on the actual traffic state and adjusts the control plan using the feedback information. The highly adaptive controllers have a high update frequency, typically within seconds instead of minutes, to quickly adjust to the actual traffic state. In general, the higher the update frequency, the more often the controller can reconsider and adapt the control plan to the actual traffic pattern.

- Control input: The controller is more adaptive if the controller gets more input, i.e., more information on the actual traffic pattern. Note that fixed-time control is based on historical demand data. Traffic-responsive control is based on real-time data on the actual traffic state, such that the controller can react on the currently measured or estimated traffic situation. The highly adaptive controllers also use predictions on future traffic states or future demands, such that the controller can pro-actively anticipate on upcoming traffic situations, which is called *predictive control*. In general, the more control inputs, the more information the controller has available to react to, and to anticipate on the actual and upcoming traffic pattern.
- Control freedom: The controller is more adaptive if the controller has a higher degree of freedom, i.e., more control actions to choose from. Note that in fixed-time control a fixed cyclic control plan is used which is predetermined offline. In traffic-responsive control the basic cyclic control plan is adapted online, for example by optimizing cycle time and green splits. The highly adaptive controllers also optimize the order and combination in which the different movements get green, letting go the idea of imposed cycles, which is called *structure-free control*. In general, the more control freedom, the more options the controller has to adapt to the actual traffic pattern.

In this thesis the focus is on highly adaptive control, i.e., on structure-free predictive feedback control systems with a high update frequency, which can quickly anticipate on the upcoming demand pattern up to the level of individual vehicle movements. Such a highly adaptive traffic signal control system has in theory a large potential to increase control performance in a traffic network. The choice and quality of the predictive component is important for the controllers' adaptivity and performance level. Therefore, in the next section, existing short-term prediction methods and their applications in traffic signal control are described.

1.1.2 Prediction methods

Since the beginning of this century, traffic prediction has been a rising topic and many traffic forecast methods have been developed, following different methodological approaches. An overview and classification of used methodologies for short-term traffic forecasting is given in [10], divided in naive, parametric, and non-parametric approaches, and the advantages and disadvantages of these methods are addressed. The non-parametric approaches have been given much attention in recent literature. An extensive review of these non-parametric methods can be found in [13], [9], [14], in which this development is closely followed over the years, and the potential of these methods is underlined. According [10]:

- Naive approaches. In naive approaches, the model structure and parameters are fixed, and are not adapted based on data. To this class of approaches belong for example statistical interpolation techniques.
- Parametric approaches. In parametric (or model-based) approaches, the model structure is fixed, and the parameters are adjusted based on data. The model relations are explicitly described as a white-box. To this class of methods belong analytical approaches, e.g., queuing methods and store-and-forward models, macroscopic traffic flow models, and meso- or microscopic simulation models (see [15], [16], [17] for traffic models in general).

In these parametric approaches, data assimilation techniques, like Kalman filters or particle filtering, are often used to fit the outcomes of the models to real-time data.

- Non-parametric approaches. In non-parametric (or data-driven) approaches, the model structure and values of model parameters are determined from data. The model relations are considered as a black-box, and the model structure and its parameters are derived by training from data. To this class of approaches belong statistical modeling, e.g., linear regression models and linear(ized) problems in state-space form [18], and machine learning techniques, e.g., neural networks and other non-linear regression models [13].

Naive methods are fast, scalable, and easy to implement, as stated by [10], however the prediction accuracy of these methods is low unless they are combined with more sophisticated methods. Non-parametric methods can make more accurate predictions, have low runtimes (once trained), and are able to capture the traffic dynamics, especially the non-linear machine learning techniques [13]. However, non-parametric methods are a location-specific black-box and can only predict situations that are present in the data. The methods are trained for a specific situation and location, with the risk of over-training, and are not easily transferable to a network context. Parametric methods have a network-wide application and, as a white-box, have the explanatory power to predict the effects of non-recurrent events. However, the calibration of these model-based methods, especially for simulation models, is an underdetermined problem, which easily leads to overfitting. Hybrid approaches, combining parametric and non-parametric methods, are indicated by [10], and [9], [14], as a promising research direction, in which the explanatory power of parametric models is combined with the accuracy and low runtimes of machine learning techniques.

In traffic signal control with a predictive component, mostly parametric methods are used, speaking of model-based control [19]. A model, with explanatory power, is used to predict the effects of (candidate) control actions. This can be done based on analytical traffic flow models, for example store-and-forward models [20], [21], or by more advanced macroscopic traffic flow models, like the link transmission model [22]. Microscopic simulation models are hardly used in relation to predictive traffic signal control, due to the high computation times. Non-parametric methods are not used much yet in predictive traffic signal control, due to the lack of explanatory power to determine the effects of the control actions. However, there are control methods that replace part of the flow modeling that is complex or computational expensive, like the queuing model, by a non-parametric approach as a self-learning black-box, as was already proposed by [23]. The improvement of the, mostly model-based, predictive component in traffic signal control is still an ongoing process and a main effort in recent literature [24].

1.1.3 Prediction uncertainties

In this thesis the focus is on model predictive traffic signal control [24], a sub-class of model-based control [19]. A prediction model is used to evaluate the effects of candidate control plans on the future traffic states in a network, and the control plan is chosen that explicitly optimizes the performance of the traffic system (see [24], [19], and [25], [26], [27] for model predictive control in general). Figure 1.1 outlines the closed-loop process of model predictive traffic signal control. The traffic process in the real world evolves over time, based on the internal traffic relations and external disturbances (e.g., demand, route choices). The traffic process is monitored by sensors (e.g., detectors), and the current traffic state (e.g., queues) and disturbances (e.g., demand) are estimated based on these observations. The controller uses the current traffic state as a starting point to determine a control plan. A prediction model evaluates the effect of candidate control plans by predicting the future states of the traffic system based on the current traffic state and the estimated disturbances. The control optimizer compares the evaluated control plans and selects the control plan that optimizes the performance of the traffic system according to a pre-defined control objective. The traffic signals of the implemented control plan influence the traffic process, and observations on the changed traffic state are used to update the prediction and the control plan. The control process repeats itself following a rolling horizon approach, shifting the prediction horizon and reinitializing to the actual traffic state, after which the control plan is adapted.

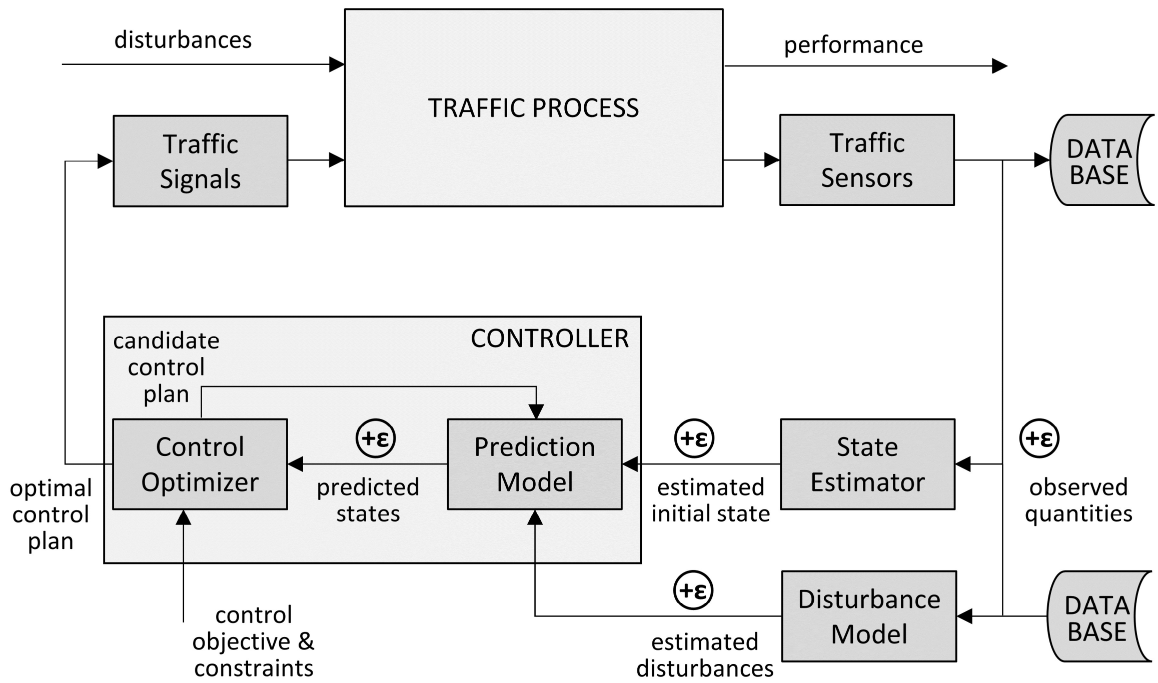


Figure 1.1: Predictive traffic signal controller with prediction errors ε.

Predictive traffic signal controllers anticipate on future traffic conditions and consider the future effects of the possible control actions, even if these effects, e.g., in the case of spillback, are not immediately visible due to longer response times of the traffic dynamics. Therefore, predictive controllers have the potential to improve control performance considerably, compared to non-predictive traffic-responsive controllers that react to the current traffic state only. This applies especially to the highly adaptive predictive controllers that reconsider control decisions frequently and have a large degree of freedom to adapt control decisions matching upcoming individual vehicle patterns. However, these highly adaptive predictive controllers depend on more, correct, and detailed information on future traffic conditions. Errors in the predicted traffic conditions, as appear in real life, may reduce the performance benefit of these systems.

Figure 1.1 shows the presence of errors in the predictive traffic signal control system. Note that the estimated initial traffic state (e.g., queues) already may contain errors, either introduced in the estimation method or already present in the underlying observations. The estimated input disturbances (e.g., demand) may also contain errors, caused by the disturbance model or inaccurate observations. In addition, the prediction model itself introduces errors in the control system, predicting the evolution of the traffic process as would appear in real life, deviating more as the prediction horizon increases and the traffic condition becomes more uncertain. The resulting errors in the predicted states are a combination of the errors in the total prediction process, from the input errors in the underlying observations and estimations to the errors in the prediction model itself. These total prediction errors may influence the control decision made by the optimizer, eventually decreasing the control performance.

However, existing literature on predictive traffic signal control mostly assumes that the proposed prediction method is perfect, whereas in real life prediction errors are present in the control system (see Figure 1.1). Until now, there has been very little study of the effect of data, estimation, and prediction errors on the performance of the traffic signal control system. This gap was already indicated for traffic management systems in general by [28], and for model-based predictive traffic signal control systems in particular by [29] and [30], [24]. Whereas, for designing and implementing these predictive traffic signal control systems, it is important to know which prediction errors cause the most performance decrease and which improvements in prediction accuracy will lead to the largest performance increase. Therefore, this thesis analyzes the impact of prediction errors on the control performance of model-based predictive traffic signal control and, vice versa, studies the performance benefit of more accurate prediction methods in predictive traffic signal control systems. Besides, methods are explored to explicitly incorporate prediction uncertainties in the control design to reduce the effect of prediction uncertainties on the control performance, especially in cases where prediction methods are not accurate enough and influence performance significantly.

1.1.4 Robust traffic signal control

In the design of traffic signal control systems, uncertainties are not incorporated much yet. The last two decades, the attention for this topic has been slightly increasing. In fixed-time control with offline predefined control plans, robust control techniques have been introduced that explicitly consider uncertainties in the disturbances of the control system (for an overview see [31] and [32]). In this context, robust approaches are used to protect against high performance loss due to large day-to-day demand fluctuations, that were not considered in the original control design based on the average demand pattern. These robust fixed-time traffic signal controllers are developed as an alternative for adaptive control methods, that need a more complex infrastructure to detect the actual demand pattern and adapt the traffic signals accordingly. In the fixed-time traffic signal controllers, robust approaches are used instead to build in margins in the offline control plan such that, once the control plan is active, a deviation from the average demand pattern can still be controlled adequately, preventing a worst-case scenario with unforeseen and unproportionally large performance loss. In the last two decades, different robust fixed-time controllers have been designed including enough margin in the control plan to protect against performance loss in these worst-case scenarios, without becoming overprotective, i.e., without decreasing the control performance too much for the average demand pattern (see [31], [32]).

In the robust fixed-time controllers, different approaches can be distinguished to explicitly consider demand uncertainties in the control design (see [32]), all resulting in a more robust control plan that reduces large performance loss due to demand fluctuations. (A more detailed classification is given in Chapter 4.) Note that the basis of the approaches is an offline optimization of the fixed-time (cyclic) control plan, optimizing the average delay that corresponds to the average demand pattern. The robust approaches differ in the way demand uncertainties are included in the optimization process.

- Distribution-based approach: Including knowledge on the demand distribution in the optimization process, optimizing a combination of the expected value of the mean and variance of the corresponding delay. The complete demand distribution is assumed to be known. A distribution-based control optimization reduces the tail of the distribution of the control performance, reducing the chance on large performance loss.
- Scenario-based approach: Including knowledge on demand scenarios with probabilities in the optimization process, optimizing the delay averaging over all scenarios (expected mean), the delay of the high-risk scenarios, or only the delay of the worst-case scenario. Scenarios with discrete probabilities are assumed to be known, or scenarios are drawn from a distribution following a stochastic programming approach. A scenario-based control optimization reduces the change on a scenario with large performance loss.
- Boundary-based approach: Including knowledge on the demand boundaries in the optimization process, optimizing the delay of the worst-case scenario. A robust (min-max) mathematical programming approach is applied based on an area formed by the demand boundaries (see [33] for robust optimization in general). Only the boundaries of the demands need to be known. A robust (min-max) control optimization guarantees an upper bound for the control performance under uncertainties.

Similar robust control approaches can also be found in adaptive traffic signal control, but the existing literature is limited. Control optimization under uncertainties is a time-consuming process, which becomes more difficult to apply in real time. So far, only a few approaches fully determine the robust signal control program online in real time, mainly applied in a model predictive control system considering cyclic control (see major recent approaches [29], [34] and [30]). Note that adaptive control, unlike fixed-time control, already anticipates to large day-to-day demand fluctuations by design. Existing robust adaptive (predictive) control methods therefore protect against performance loss due to smaller time-varying (mostly cyclic) demand uncertainties left in the system. If predictive control systems become more adaptive with more degrees of control freedom (no imposed cycles), demand fluctuations are anticipated further by design up to the individual vehicle level, and other types of uncertainties that are present in de prediction model itself become more dominant in the system performance. Existing robust approaches do not yet consider these kind of model uncertainties and resulting prediction errors in these highly adaptive (structure-free) control systems. Therefore, this thesis studies robust control principles in a real-time adaptive context and analyzes whether robust techniques in highly adaptive (structure-free) model-based predictive signal control protect against performance loss due to uncertainties in the prediction model.

1.2 Research objective

Considering prediction uncertainties that influence the control performance of highly adaptive predictive traffic signal control systems, the main research objective of this thesis is twofold and can be formulated as follows:

1. Prediction quality: *To analyze the effect of prediction errors on the control performance of highly adaptive (structure-free) model-based predictive traffic signal control systems* [Chapter 2, 3].

2. Control robustness: *To design a robust control approach that explicitly considers prediction uncertainties in highly adaptive (structure-free) model-based predictive traffic signal control systems to reduce the effect of prediction errors on the control performance* [Chapter 4].

The research objectives are addressed in a real-time adaptive control context, considering additional requirements on real-time applicability. The insights into prediction quality and the robust control approach should be usable for the design of highly adaptive (structure-free) model-based predictive traffic signal control applications in real life [Chapter 5].

1.3 Research questions

Until now, the relation between prediction quality (accuracy) and control performance has been barely analyzed, and control robustness has been hardly considered in predictive traffic signal control. Therefore, in this thesis the following research questions are addressed, regarding prediction quality (objective 1) and control robustness (objective 2):

1A. *What is the relation between prediction errors and control performance of a highly adaptive (structure-free) model-based predictive traffic signal control system for a local-controlled single intersection?* [Chapter 2]

Prediction errors in the different input and output quantities of the control system on a single intersection are considered and the local effects on the control performance at the intersection are analyzed. Control systems with different adaptivity levels are compared, regarding the prediction horizon length (control input), update frequency (control update), and degrees of freedom (control freedom), with a focus on the control freedom comparing structure-free control to the more traditional cyclic controllers. The obtained insight is used to define guidelines for the design of real-life predictive traffic signal control systems under uncertainties, regarding the prediction horizon length, update frequency, degree of control freedom, and the most important quantities to predict accurately.

1B. *What is the relation between prediction errors and control performance of a highly adaptive (structure-free) model-based predictive traffic signal control system for network controlled multiple intersections?* [Chapter 3]

Prediction errors in the different model quantities of the control system on a network are considered, and the global effects on the control performance in the network are analyzed. In addition to 1A, the error propagation in the network is addressed, considering traffic flow relations between the quantities of different intersections in forwards (driving) and backwards (spillback) direction. As in 1A, different adaptivity levels are analyzed, but now with a focus on control input comparing local intersection control and global network control, increasing the prediction horizon length looking ahead over multiple intersections. Note that the problem complexity of (structure-free) model-based predictive traffic signal control increases rapidly with the number of additional intersections. Model aggregation is often applied to speed up the calculation process of the prediction model. Therefore, in addition to 1A, prediction errors are distinguished in aggregation errors and additional biases, to analyze the effect on the control performance of both types of prediction errors. Besides, a heuristic control method is designed to speed up the optimization process, and the effect of suboptimal control decisions on the control performance is monitored. The obtained insights are used to extend the guidelines, as defined in 1A, for the design of real-life predictive traffic signal control systems under uncertainties on a network level.

2. *To what extent can robust control techniques, applied in a real-time adaptive context, reduce the effect of prediction errors on the control performance due to parameter uncertainties in highly adaptive (structure-free) model-based predictive traffic signal control?* [Chapter 4]

Considering uncertainties in the prediction model parameters (or quantities) that cause the largest performance loss in the predictive network control system, as analyzed in 1B, robust approaches are explored to reduce the performance loss due to these prediction uncertainties.
The benefits of robust control techniques to protect against large performance loss due to parameter uncertainties in the predictive network control system are analyzed, as well as the drawbacks of robust control approaches, like overprotectiveness (see Section 1.1.4). Note that the problem complexity of the predictive network controller, as analyzed in 1B, increases further after including uncertainties. Heuristic robust approaches are explored to reduce calculation times of robust control techniques. The obtained insights are used to design a (heuristic) robust control method that explicitly considers parameter uncertainties in (structure-free) model-based predictive traffic signal control systems, and is suitable for a real-time adaptive context.

The research questions are addressed in a real-time adaptive control context, for which the predictive traffic signal control system is eventually designed. Requirements on real-time applicability are already considered in the research process, making the design step to real-life predictive traffic signal control applications smaller. A ready-to-use testbed for advanced real-time adaptive predictive traffic signal control is not available, therefore a prototyping environment is set up during this research [Chapter 5].

1.4 Research approach

A prototyping environment [Chapter 5] is set up to study real-time adaptive (structure-free) model-based predictive traffic signal controllers under prediction uncertainties. The environment is based on an existing microscopic simulator running in real time, in which adaptive control systems function on the same individual vehicle level with the same timing as in real life, but still run in a controlled environment where experiments can be structurally executed. A real-time adaptive (structure-free) model-based predictive traffic signal controller is designed on top of the simulator, requiring a heuristic approach to make the real-time predictive controller applicable on a network. The effect of prediction errors on the control performance is studied in the prototyping environment, and robust approaches are explored to reduce the effect of prediction uncertainties. During the design of the predictive control system and its robust extension, requirements are set regarding real-time applicability of the controller. The requirements are addressed during the research process, providing a suitable implementation to run the experiments. In the end, a working prototyping environment for real-time (robust) predictive control systems is obtained, which can be used for future research. The insights obtained in the real-time prototyping environment can be used for the development and design of predictive control systems in real life.

In the prototyping environment two different types of research are performed in this thesis:

1. Sensitivity analysis. [Chapter 2, 3]
   To address research questions 1A,B, a sensitivity analysis is performed for the real-time adaptive (structure-free) model-based predictive traffic signal control system to study the effect of prediction errors on the control performance. The ideal world is created with perfect predictions for the predictive control system in the microscopic simulation environment. Note that the ideal world with perfect predictions can only be created in a controlled simulation environment, where future traffic behavior is known, and not in real life, where the future traffic evolution is uncertain. In the ideal simulated world, the potential behavior and performance of the control system is analyzed. Then, prediction errors are introduced in a structured way in the different quantities of the control system, and the performance loss due to these errors is analyzed. Note that the results of the sensitivity analysis can roughly be reversed, indicating the possible performance gain of more accurate predictions. The sensitivity analysis is first applied to local control on a single intersection in Chapter 2 and is then extended to network control on a corridor of multiple intersections in Chapter 3. The results of the sensitivity analysis are translated into design guidelines for real-life predictive traffic signal control systems under uncertainties, regarding the prediction horizon length, update frequency, degrees of control freedom, and the most important quantities to predict accurately.

2. Robust control design. [Chapter 4]
   To address research question 2, robust control approaches are studied for real-time adaptive (structure-free) model-based predictive traffic signal control systems to reduce performance loss due to prediction uncertainties. Scenario-based robust control approaches, originally developed for demand uncertainties in fixed time control (see Section 1.1.4), are translated into the real-time adaptive predictive control context to protect against performance loss

due to uncertainties in the prediction model parameters. The insights, obtained in the sensitivity analysis for (non-robust) predictive control, are used to define the worst-case scenarios where prediction errors may result in high performance loss, i.e., the most sensitive quantity in the prediction model and critical traffic situation in the corridor. The potential of the robust control approaches is analyzed on the ability to protect against performance loss in these worst-case scenarios (and on possible side effects like overprotectiveness). Besides, the problem complexity of robust control techniques is considered, and heuristic solution approaches are analyzed to reduce calculation times. The insights lead to the design of a (heuristic) robust control method that reduces the effect of prediction errors on the control performance by explicitly considering parameter uncertainties in real-time highly adaptive (structure-free) model-based predictive traffic signal control systems.

1.5 Scientific contributions

The major contribution to the field of predictive traffic signal control is twofold, consisting of an analysis on prediction quality and a robust control design. This thesis provides new insights into the relation between prediction quality (accuracy) and control performance in a predictive traffic signal control system, which can be used in the design of predictive traffic signal control applications in real life. And this thesis provides a robust control approach, that explicitly considers prediction uncertainties in the predictive traffic signal control system, proposing a new heuristic method to be applicable in a real-time adaptive context. This robust approach leads to improved predictive traffic signal control systems in which the impact of prediction uncertainties on the control performance is reduced. This thesis makes predictive signal control systems better applicable in real life, increasing control performance under prediction uncertainties.

In this subsection, the different scientific contributions to the field of predictive traffic signal control are described in more detail, distinguished in new insights into the importance of prediction quality and potential of robust approaches in predictive traffic signal control, new heuristic (robust) predictive control methods, and an open-source prototyping environment for new advanced predictive control applications.

1.5.1 Prediction quality

The main contributions regarding prediction quality are:

- A theoretical framework for sensitivity analysis into the effects of prediction errors on the control performance of predictive traffic signal control systems [Chapter 2, 3].
- New insights into the relation between prediction quality and control performance of predictive traffic signal control systems for local intersection control [Chapter 2], and network control [Chapter 3]. In both directions, insights into performance loss due to prediction errors in current control systems, as well as insights into performance gains by more accurate predictions in future predictive traffic signal control systems.

- Guidelines for the design of predictive traffic signal control systems under uncertainties, regarding the prediction horizon length, update frequency, degrees of control freedom, and the most important quantities to predict accurately [Chapter 2, 3].

1.5.2 Robust control

The main contributions regarding robust control are:

- A robust control approach for predictive traffic signal control, applying a scenario-based robust control approach in the new context of adaptive (structure-free) model-based predictive traffic signal control to protect against performance loss due to uncertainties in the prediction model parameters [Chapter 4].
- New insights into the potential of robust control techniques for predictive traffic signal control systems to reduce the impact of prediction uncertainties on the control performance [Chapter 4]. Insights into the potential of a robust control approach for adaptive (structure-free) model-based predictive traffic signal control to protect against performance loss due to uncertainties in the prediction model parameters, and insights into the drawbacks of robust control in such an adaptive predictive control context (like overprotectiveness and computational complexity).

1.5.3 Heuristic control methods

The main contributions regarding heuristic control methods are:

- A new heuristic optimization approach for structure-free model-based predictive traffic signal control to be applied in a real-time adaptive context [Chapter 3].
- A new heuristic robust control method considering uncertainties in prediction model parameters in structure-free model-based predictive traffic signal control to be applied in a real-time adaptive context [Chapter 4].

1.5.4 Real-time prototyping

The main contributions regarding real-time prototyping are:

- An open-source implementation of a prototyping environment for real-time adaptive predictive traffic signal control applications, which is made available for future research [Chapter 5]. A prototyping environment to study new types of advanced real-time highly adaptive predictive traffic signal control applications, to obtain valuable insights for the design in real life.
- New insights into the potential of new types of advanced real-time highly adaptive predictive traffic signal control applications, obtained in the prototyping environment, like the potential of highly adaptive structure-free controllers in comparison to the more traditional cyclic predictive control systems [Chapter 2], and the potential of including individual vehicle information in predictive traffic signal control systems [Chapter 3].

1.6 Thesis outline

Figure 1.2 presents the outline of this thesis on the impact of uncertainties in predictive traffic signal control. The thesis addresses the two main research objectives on prediction quality and control robustness (see Section 1.2). In Chapter 2 and 3, a sensitivity analysis is described to analyze the relation between prediction quality and control performance for predictive traffic signal control. In Chapter 4, a robust approach is described to explicitly consider prediction uncertainties in the design of predictive traffic signal control. Chapter 5 is meant as a supporting chapter, describing the real-time prototyping environment for predictive traffic signal control systems, which is used throughout the entire thesis.

In Chapter 2 and 3 the sensitivity analysis is described to analyze the impact of prediction errors on the control performance of predictive traffic signal control. First, in Chapter 2, a predictive traffic signal control system is designed for local intersection control. A sensitivity analysis is performed to study the effect of prediction errors on the control performance on a single intersection. Then, in Chapter 3, the predictive traffic signal control system is extended for network control. A sensitivity analysis is performed to study the effect of prediction errors on the control performance on a corridor of multiple intersections. In both Chapters 2 and 3, the results of the sensitivity analysis are translated into design guidelines for predictive traffic signal control systems under uncertainties.

In Chapter 4 a robust approach is described to reduce the impact of prediction uncertainties on the control performance of predictive traffic signal control. The predictive traffic signal control system for network control of Chapter 3 is extended to explicitly consider multiple uncertainty scenarios. The insights from Chapter 3 into the prediction errors with the largest impact on the control performance on the analyzed corridor is used to define the uncertainty scenarios. In Chapter 4, the robust approach is analyzed on the corridor on the ability to reduce the impact of these prediction uncertainties on the control performance of the robust predictive control system.

In Chapter 5 the real-time prototyping environment is described that is used throughout the entire thesis. The prototyping environment is set up during the research. In the design of the predictive control system in Chapter 2 and 3 from local to network control, and in the design of its robust extension in Chapter 4, requirements are set regarding real-time applicability of the controller. These requirements and implementations are described in Chapter 5, resulting in a complete prototyping environment for real-time (robust) predictive control systems, that can be used in future research. The prototyping environment is used to perform the sensitivity analysis in Chapter 2 and 3, and the robust analysis in Chapter 4, providing the valuable insights in this thesis into the importance of prediction quality and potential of control robustness in predictive traffic signal control.

In Chapter 6 the results are translated into conclusions and recommendations for future research, and practical implications for the design of predictive traffic signal control applications in real life.

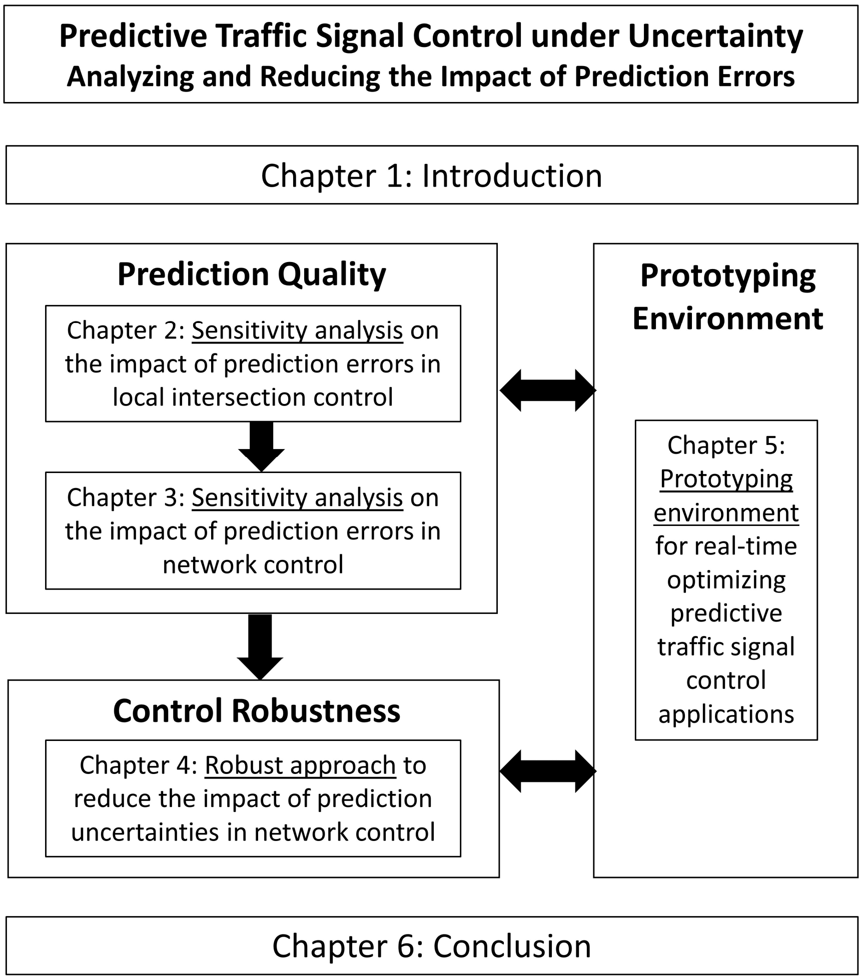


Figure 1.2: Thesis outline.

Chapter 2

Sensitivity analysis on the impact of prediction errors in local intersection control

Signalized traffic control is important in traffic management to reduce congestion in urban areas. With recent technological developments, more data have become available to the controllers and advanced state estimation and prediction methods have been developed that use these data. To fully benefit from these techniques in the design of signalized traffic controllers, it is important to look at the quality of the estimated and predicted input quantities in relation to the performance of the controllers. Therefore, in this chapter, a general framework for sensitivity analysis is proposed, to analyze the effect of erroneous input quantities on the performance of different types of signalized traffic control. The framework is illustrated for predictive control with different adaptivity levels. Experimental relations between the performance of the control system and the prediction horizon are obtained for perfect and erroneous predictions. The results show that prediction improves the performance of a signalized traffic controller, even in most of the cases with erroneous input data. Moreover, controllers with high adaptivity seem to outperform controllers with low adaptivity, under both perfect and erroneous predictions. The outcome of the sensitivity analysis contributes to understanding the relations between information quality and performance of signalized traffic control. In the design phase of a controller, this insight can be used to make choices on the length of the prediction horizon, the level of adaptivity of the controller, the representativeness of the objective of the control system, and the input quantities that need to be estimated and predicted the most accurately.

This chapter is based on the paper: M. C. Poelman, A. Hegyi, A. Verbraeck, and J. W. C. van Lint, "Sensitivity Analysis to Define Guidelines for Predictive Control Design," *Transp. Res. Rec.*, vol. 2674, no. 6, pp. 385–398, 2020. https://doi.org/10.1177/0361198120919114

This chapter is also presented as poster at *TRB 99th Annual Meeting*, Washington D.C., 2020.

A preliminary version of this research is presented at *hEART*, Budapest, 2019, entitled: M. C. Poelman, A. Hegyi, A. Verbraeck, and J. W. C. van Lint, "Sensitivity Analysis on Information Quality for Signalized Traffic Control," *8th Symp. Eur. Assoc. Res. Transp.*, 2019.

2.1 Introduction

Signalized traffic control is important in traffic management to reduce congestion in urban areas. With recent technological developments, more data have become available to the controllers, varying from historical to real-time data, and from location-based data (like loop detectors) to floating-car data. Advanced state estimation and prediction methods have been developed that use these data [9], [10]. Some of these methods have already been applied in controllers to optimize traffic conditions proactively [11], [5]. To benefit fully from these techniques in signalized traffic controllers, it is important to look at the quality of the estimated and predicted input quantities in relation to the performance of the controllers. For the development of estimation and prediction methods on the one hand, and the design of traffic controllers on the other, it is important to have insight into the extent to which the accuracy of estimation and prediction will affect the performance of the controller.

Therefore, in this chapter, the sensitivity of signalized traffic control for erroneous input quantities is addressed. A general framework for sensitivity analysis is proposed, to analyze the effect of errors in the measured, estimated and predicted input quantities on the performance of different types of signalized traffic control. The framework is applied to predictive control, to analyze to what extent a prediction increases the performance of the controller, considering that the prediction contains errors. The results of the sensitivity analysis framework are used to set up design guidelines for predictive control.

In this chapter, the following section gives a problem description followed by a short discussion of the state of the art on traffic control and sensitivity to input quantities. Then the sensitivity analysis framework is outlined and demonstrated for predictive control with different adaptivity levels. The results of the sensitivity analysis are presented and translated into design guidelines, and the chapter is concluded with directions for future research.

2.2 Problem description

In Figure 2.1, the process of signalized traffic control is outlined in relation to control theory. The controller influences the traffic process by its control signal. The traffic process is evolving in time, based on the internal traffic relations and external disturbances (demand, route choices). The traffic process can be monitored in real time by sensors, resulting in observed quantities. The observed quantities can be used to estimate the actual state of the traffic system expressed in derived quantities (like queue lengths). Likewise, the observed quantities can also be used to estimate (and predict) the disturbances. Based on the estimated state of the traffic system and a prediction of the disturbances, the future state of the traffic system can be predicted. Information on historical, actual and future traffic states (combined with information on disturbances) is used as input for the controller. Based on this information, the controller determines the control scheme that implicitly or explicitly optimizes the performance of the traffic system.

In this control process, errors may arise that can influence the control decision. In general, errors in the input quantities will eventually decrease the performance of the controller. Therefore, it is important to look at all elements in the control process where errors may occur. In monitoring the traffic system, an observation error will occur, caused by the inaccuracy of the sensor and observation method that is used. In the estimation of the traffic state (and disturbances) an estimation error is introduced, which may represent errors introduced by the estimation method itself or errors that were already present in the observed quantities. In the prediction of the traffic state (and disturbances), a prediction error is introduced. This error depends on the original error of the estimated state, and the prediction method. This prediction error will increase with the prediction horizon.

In the design of the controller as well as the estimation and prediction methods, it is important to know to what extent these errors influence the control decision and the performance of the controller. In this chapter, this question is addressed by proposing a framework for sensitivity analysis on the observed, estimated, and predicted input quantities of signalized traffic control.

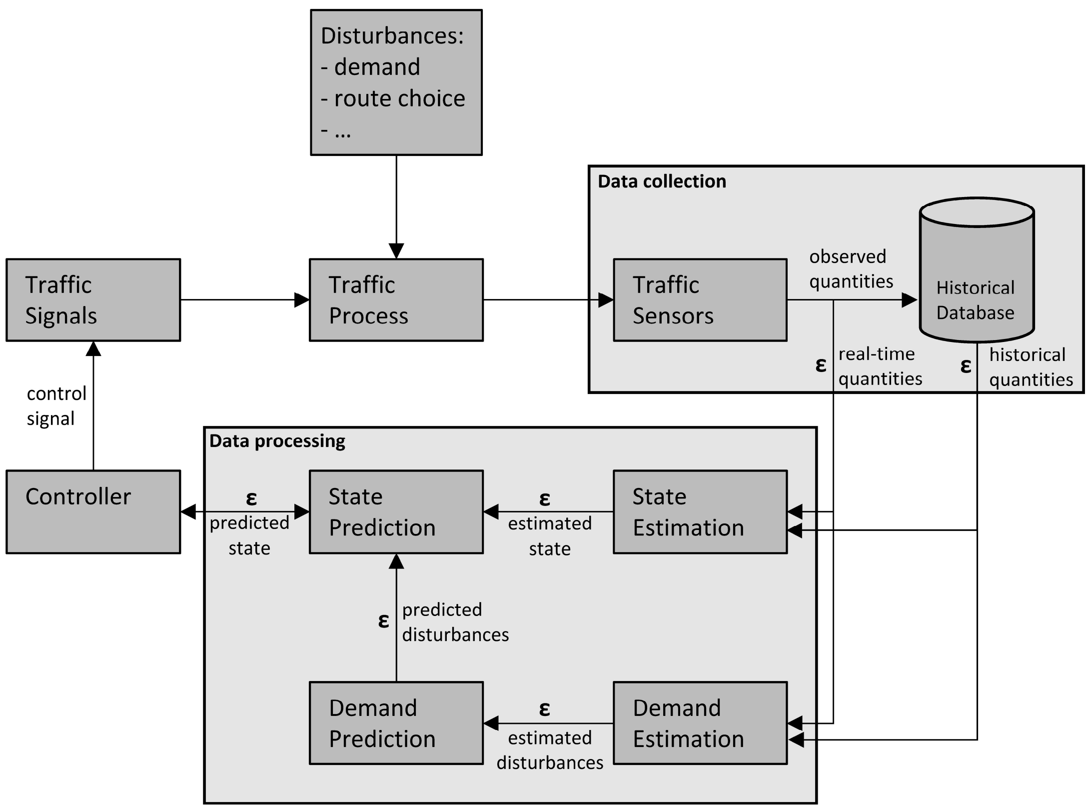


Figure 2.1: Process scheme of signalized traffic control with errors (ε) arising in the control process.

2.3 State of the art

There is a wide variety of types of signalized traffic control [11], [5], [1]. Signalized traffic control methods can be divided into two general categories: fixed-time control and traffic-responsive control. In fixed-time control, the control is optimized off-line, based on historical demand data. In traffic-responsive (or adaptive) control, the control is adapted in real time based on on-line data. The controller can react to the currently measured or estimated traffic situation, or it can proactively anticipate predicted traffic conditions. In general, the more detailed information is used, the more sensitive the controller performance is expected to be for errors in this information. Fixed-time control is quite robust for errors in the input quantities containing margins by design (implicitly in Webster-based cycle times, explicitly in robust control [32]). Traffic-responsive (or adaptive) control will be more sensitive to information errors, depending on the degrees of freedom of the controller.

Different levels of adaptive control can be distinguished by the degrees of freedom in the controller [11]. In the first level, predefined control schemes are selected from a library based on the actual traffic conditions. In the second level, the control schemes are assumed to be cyclic, and cyclic parameters (like green splits) are adapted, based on information about the traffic conditions for current and upcoming cycles. In the third level, the control scheme is considered structure-free (no cycles). The combination and the order of movements can be adapted, together with the green times. In general, the more degrees of freedom there are in the controller, the better performance can be reached, the more sensitive the controller will likely be for errors in the estimated or predicted traffic conditions. In this chapter this sensitivity will be analyzed for controllers with different degrees of freedom, varying from cyclic to structure-free control.

In the field of traffic management, sensitivity analysis on information errors has not yet received much attention. This holds not only for signalized control, but also for dynamic traffic management in general [28]. The attention to this kind of analysis seems to have increased because of the introduction of floating-car data in the field of dynamic traffic management. For signalized control based on floating-car data, evaluating the influence of penetration rates and additional data errors is essential for a well-functioning system [35]. With the increase of adaptivity of traffic controllers and the availability of more detailed information, the need for these sensitivity analyses on information errors is still increasing. This chapter contributes to research on this issue.

2.4 Experimental framework

In this section, a general framework for sensitivity analysis is proposed to analyze the effect of errors in the measured, estimated and predicted input quantities on the performance of different types of signalized traffic control. Assuming perfect information, the ideal situation for a signalized traffic controller is created. Perfect information can be perfectly observed historical or real-time data, a perfect state estimation, or perfect prediction (no errors). Using a Monte Carlo approach, the perfect information is randomly disturbed, and the degeneration in performance of the controller is evaluated. The outcome of the sensitivity analysis will be an experimental relation between the level of information quality and the performance of signalized traffic control in the traffic system. The experimental framework is outlined in Figure 2.2.

The framework makes use of a simulation environment to represent the real world. In an additional Application Programming Interface (API) the controller of interest is interacting with the simulation environment. A network configuration and a relevant demand scenario is chosen. Since the main goal of the sensitivity analysis is to determine the effect of errors in the input quantities of a controller and not to determine the effects of fluctuations in demand, the realization of the demand pattern is fixed during the sensitivity analysis. However, the sensitivity analysis can be repeated for different demand patterns (and network configurations) to compare the sensitivity for control input errors in different traffic conditions.

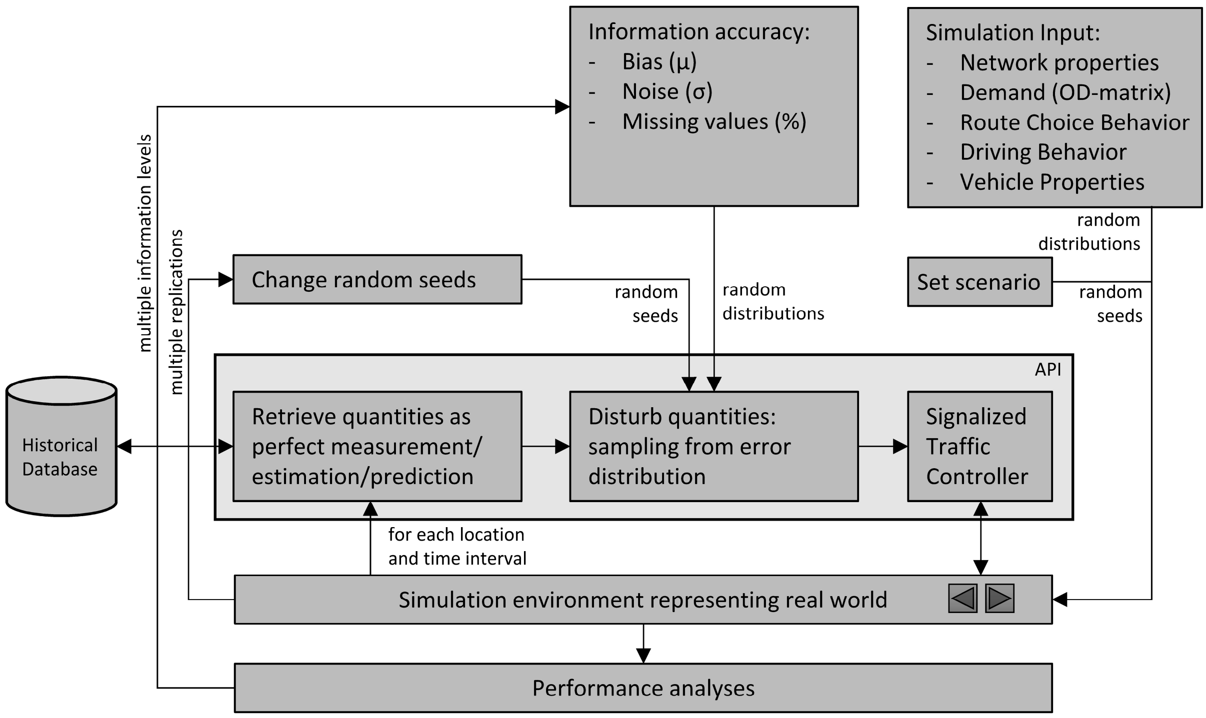


Figure 2.2: Experimental Framework.

The main input to the sensitivity analysis is the information quality of the input quantities for the traffic controller. Information quality may consist of many aspects. In this framework, information accuracy of the input quantities is considered, expressed in a structural bias, a random noise, and a percentage of missing data, described by a random error distribution (of a properly chosen form). It is assumed that the information accuracy depends only on the observation, estimation or prediction method, and does not depend on location and time, resulting in the same error distribution for each location and time. The realizations of the errors, however, differ over locations and time instances, and are independently drawn from the distributions. The effect of the errors can be simulated as follows:

0. Initialize the input error to no bias, no noise, no missing data (no error distribution yet) and simulate the situation with perfect information for the scenario. In this way, the ideal performance for the traffic controller is measured and set as a reference.

1. Increase the error by increasing the bias, noise, or percentage of missing vehicles. Adapt the random distributions for the control input errors accordingly.
2. Simulate multiple realizations of the errors to level out random variations over different locations and times. For each realized error pattern, for each control interval:
   2.1. Retrieve for each location the perfect input quantities from the simulation.
   2.2. Disturb the input quantities by the random realization of the error.
   2.3. Determine the control scheme based on the disturbed input quantities.
3. Measure the performance (note that it is assumed that the performance is measured perfectly in the simulation) and average over the simulated error realizations. Repeat the process from 1.

The output of the sensitivity analysis will be an experimental relation between the error in the input quantities and the performance of signalized traffic control in the traffic system for a given scenario.

2.5 Case: predictive control

The sensitivity analysis framework is in principle suitable for all types of signalized traffic control. However, in this chapter, the framework is applied to traffic-responsive control with a predictive component. The main goal of the sensitivity analysis is to analyze to what extent a prediction improves the performance of the controller, considering that the prediction contains errors. The influence of the prediction horizon is studied, assuming errors accumulate for longer horizons. Different predictive controllers are considered with increasing degrees of freedom, varying from cyclic to structure-free control, to investigate the relation between adaptivity and performance, especially under erroneous predictions. A comparison is made with non-predictive control as well. Assuming a very short prediction horizon, predictive control can be considered as non-predictive control, where the controller only reacts to the current traffic situation. This is equivalent to (conventional) vehicle-actuated control where a movement is given a green signal when vehicles are present. Since control behavior may depend on the demand, different demand scenarios are considered, i.e., undersaturated, saturated, and oversaturated conditions. The sensitivity analysis is limited to a single intersection.

In the sensitivity analysis, different relations are tested, all related to design aspects of a predictive controller. For the different demand scenarios, it is verified:

- Whether prediction improves the performance of a controller when perfect information is available. The relationship between prediction horizon and performance will be analyzed, and the prediction horizon length is identified beyond which performance does not improve any more.
- Whether prediction still improves the performance of a controller, when errors are present in the predicted input data.
- Whether a controller with high degrees of freedom, having a high adaptivity to anticipate fluctuating traffic patterns, is also more sensitive to errors in the predicted input quantities.
- Which input quantity of the controller is the most sensitive to errors and therefore the most important to estimate or predict accurately.
- Whether there are any unforeseen effects that need consideration in the design of predictive control.

In the next sections, first the predictive control model is specified in detail. Subsequently, the experimental settings of the control scenario are explained, concerning the intersection configuration, demand scenarios, and type of predictive controllers. Then the results of the sensitivity analysis are presented. Finally, the results are translated into design guidelines for predictive control.

2.5.1 Predictive control model

The sensitivity analysis framework is specified for a predictive controller for a single intersection. The basis of the framework is the micro-simulation model Aimsun (Version 8.2.0), representing the ideal world. On top of this simulation framework, a predictive controller is implemented (using API). Based on the intersection configuration, the combinations and the possible order of the movements are predefined. The free control parameters, i.e., the green times of the movements, are optimized based on a prediction of the traffic conditions.

A rolling horizon approach is used. Each control interval, the control sequence is updated in real time considering a new planning horizon. The objective of the controller is to minimize the total delay over the upcoming planning horizon, based on the current state (queues) and a prediction of the upcoming demand (arrival pattern). In the ideal simulation world, assuming perfect knowledge on the upcoming traffic situation, the expected delay could be determined by playing the simulation fast-forward for each candidate controller. To save computation time, however, a simple store-and-forward model with vertical queueing is used as a prediction model. Note that in this simplified prediction model, the predicted arrivals are perfectly known beforehand (since a single intersection is considered, the arrivals do not depend on control decisions). In the simulation environment, the non-delayed arrivals are stored and considered as the perfect predicted arrival pattern. The current state (queues) is also perfectly known from the simulation environment. Only the predicted departures are approximated by estimating vehicle passages through green, based on the state of the candidate control scheme and an approximation of the saturation flow rate. The saturation flow rate is experimentally obtained by measuring the queue discharge in the simulation environment (considering equal vehicles with the same average driving behavior).

The predictive controller can be expressed as a discrete mathematical programming problem. To this end, define the discrete time index k with duration T [s]. Let i denote the index of a movement. Define movement group with index j, as a group of non-conflicting movements that can have green at the same time, and let $I(j)$ be the set of movements belonging to movement group j. Let $J(j)$ be the set of possible movement group indexes that can follow movement group j. Introduce signal states $s_i(k) \in \{0\ (\text{red}), 1\ (\text{green})\}$ and movement group states $p_j(k) \in \{0\ (\text{red}), 1\ (\text{green})\}$, for all movements i, movement groups j, and time indexes k for the prediction horizon $[k_0, k_0+K]$. The states should satisfy a predefined combination and order of the movements, forming the constraints of the optimization problem:

Composition of movement groups is respected: $s_i(k) = p_j(k) \quad \forall i \in I(j)$ (2.1)

Exactly one movement group is active: $\sum_j p_j(k) = 1 \quad \forall k$ (2.2)

Order of the movement groups is respected: $p_{\tilde{j}}(k+1) = 0 \quad \forall \tilde{j} \notin J(j) \quad \forall j : p_j(k) = 1$ (2.3)

The objective of the controller is to find the constrained signal states $s_i(k) \in \{0,1\}\ \forall i\ \forall k$, and corresponding movement group states $p_j(k) \in \{0,1\}\ \forall j\ \forall k$, that minimize the total delay over the prediction horizon [k_0,k_0+K], i.e.,

$$\min_{\{s_i(k)\}} \sum_i \sum_{k=k_0}^{k_0+K} x_i(k) * T \tag{2.4}$$

with queue $x_i(k)$ [veh] per movement i defined as:

$$x_i(k) = x_i(k-1) + a_i(k) - d_i(k)\ \ \forall k\ \forall i \tag{2.5}$$

with arrivals $a_i(k)$ [veh] and departures $d_i(k)$ [veh] per movement i, where $d_i(k)$ is approximated by an experimentally derived saturation flow curve $r(k)$ [veh], i.e.,

$$d_i(k) = \begin{cases} \min\left(r(k), x_i(k-1) + a_i(k)\right) & \text{if } s_i(k) = 1 \text{ (green)} \\ 0 & \text{if } s_i(k) = 0 \text{ (red)} \end{cases}\ \ \forall k\ \forall i \tag{2.6}$$

Initialization: Add node n_0 with current state $s_i^{(n_0)}(k_0)\ \forall i$, $p_j^{(n_0)}(k_0)\ \forall j$, queue $x_i^{(n_0)}(k_0)\ \forall i$, and delay costs $c^{(n_0)}{}_{[k_0]} = 0$ to the search-list. Set time k=k_0, minimum costs $C_{MIN} = \infty$, and initialize greedy solution $n_{\text{GREEDY}} = n_0$

While the search-list is not empty do:

1. If available, select and remove n_{GREEDY} from the search list, set $n = n_{\text{GREEDY}}$, reset k to the time index of node n
   Else, select and remove the first node n from the search list, and reset k to the time index of node n

2. If the end of the planning horizon is reached, i.e., k=k_0+K, check the optimum:
   If $c^{(n)}{}_{[k_0,k_0+K]} < C_{MIN}$, store the optimal sequence $p_j^{(n_0)}(k_0), \ldots, p_j^{(n)}(k_0+K)$ and set $C_{MIN} = c^{(n)}{}_{[k_0,k_0+K]}$

3. Else, reset n_{GREEDY} = NONE, $C_{\text{GREEDY}} = \infty$, and branch the node n at time index k to the next time index k+1
   For each possible movement group transition $j^{\sim} \in J(j)$, from current movement group $j\!: p_j^{(n)}(k) = 1$, do

   3.1. Define a new node m with state $s_i^{(m)}(k+1) = p_{j^{\sim}}^{(m)}(k+1) = 1\ \ \forall i \in I(j^{\sim})$, and set BOUND = FALSE
   3.2. Update queues $x_i^{(m)}(k+1)$ by Eq. (2.5), and delay $c^{(m)}{}_{[k_0,k+1]} = c^{(n)}{}_{[k_0,k]} + \sum_i x_i^{(m)}(k+1) * T$
   3.3. Determine an underestimation of the remaining costs till the end of the horizon, $\tilde{c}^{(m)}{}_{(k+1,k_0+K]}$

   3.4. If state $s_i^{(m)}(k+1)$ violates control constraints (min/max green-times), then BOUND = TRUE
   3.5. If state $s_i^{(m)}(k+1)$ is already present in the decision tree, then BOUND = TRUE, i.e., $\exists\ m^{\sim}$at k+1:
   $s_i^{(m^{\sim})}(k+1) = s_i^{(m)}(k+1)\ \forall i,\ x_i^{(m^{\sim})}(k+1) \le x_i^{(m)}(k+1)\ \forall i$, and $c^{(m^{\sim})}{}_{[k_0,k+1]} \le c^{(m)}{}_{[k_0,k+1]}$
   3.6. If delay $c^{(m)}{}_{[k_0,k+1]} > C_{MIN}$, then BOUND = TRUE
   3.7. If underestimation of delay $c^{(m)}{}_{[k_0,k+1]} + \tilde{c}^{(m)}{}_{(k+1,k_0+K]} > C_{MIN}$, then BOUND = TRUE

   3.8. If NOT BOUND, then:
   Add node m to the search list (sorted on underestimation of delay)
   Store the greedy node, i.e., if $c^{(m)}{}_{[k_0,k+1]} < C_{\text{GREEDY}}$, then set $n_{\text{GREEDY}} = m$, $C_{\text{GREEDY}} = c^{(m)}{}_{[k_0,k+1]}$

Figure 2.3: Pseudo code of branch-and-bound algorithm.

The discrete mathematical programming problem is solved following a branch-and-bound approach using decision trees. The pseudo code of the branch-and-bound process is given in Figure 2.3. Each node n in the decision tree is formed by the signal states $s_i^{(n)}(k)$ and corresponding movement group states $p_j^{(n)}(k)$ at time k, starting in the current state at the beginning of the planning horizon, $k=k_0$ (Initialization). The most promising node of the decision tree is selected to expand (Step 1). If the end of the planning horizon is reached, the sequence of movement groups is checked for its optimality (Step 2). While the end of the planning horizon is not yet reached, the node is branched to the next time interval of the planning horizon (Step 3). To this end, for each possible movement group transition, the new signal states are calculated (Step 3.1), and the queues and delay are updated (Step 3.2). Based on this new state information, it is decided if the new node is added to the search tree (branched) or is discarded (bounded). It is checked whether the state violates additional control constraints (Step 3.4), whether the state is already present in the decision tree with comparable or lower delay (Step 3.5), whether the delay is larger than the minimum delay so far (Step 3.6), if so, the node is discarded, otherwise, it is branched. Branched nodes are added to a search list (Step 3.8), from which the algorithm can continue the search process (Step 1).

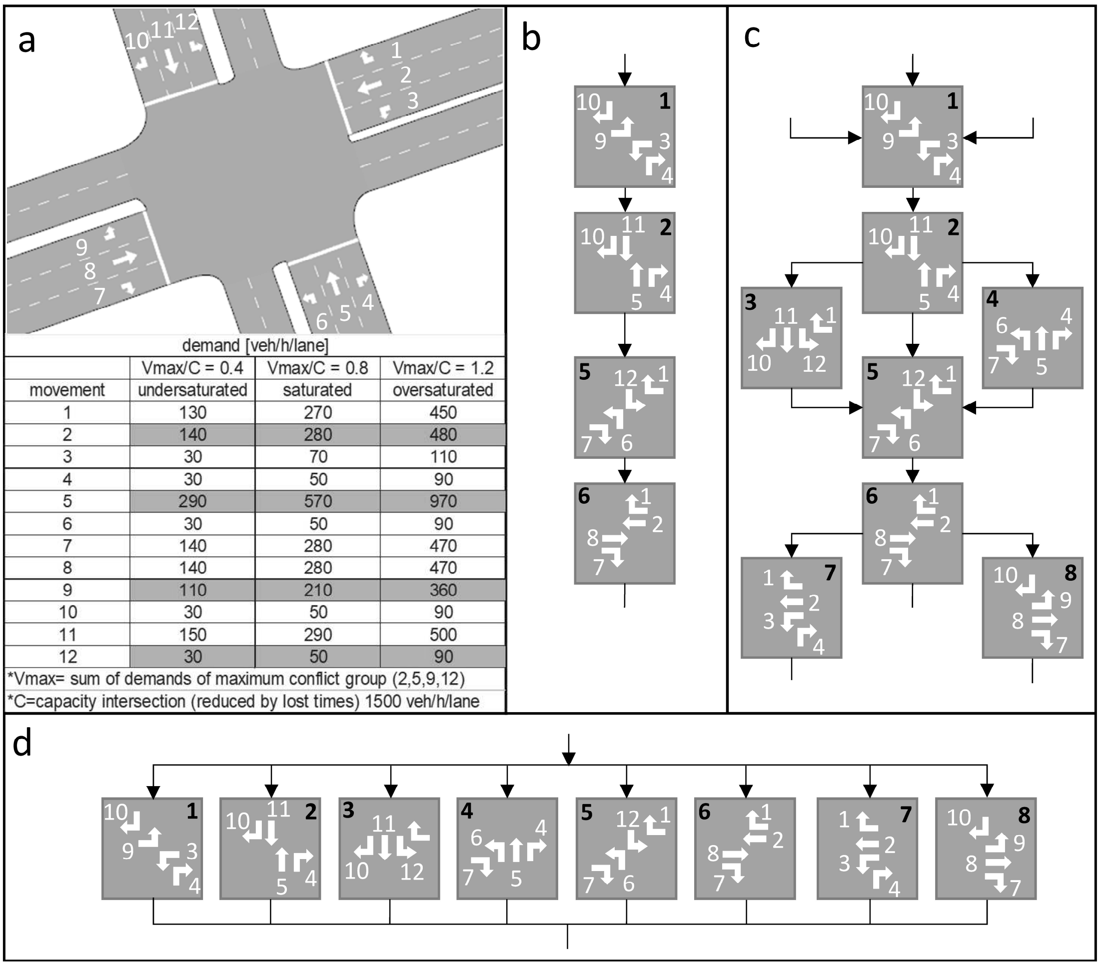
The depth of the decision tree is determined by the length of the prediction horizon, the width of the tree by the possible movement group transitions. The width of the decision tree can become quite large for increasing prediction horizons, especially if the set of movement group transitions $J(j)$ is large. To be able to solve in real time, additional bound criteria are introduced that limit the size of the search space but still guarantee optimality. When the state and delay of a new node is updated, an underestimation of the entire delay to the end of the planning horizon is made (Step 3.3). This underestimation is used to check if the node can already be bounded (Step 3.7). Moreover, greedy initial solutions are used to speed up the search process. If a node is added (Step 3.8), and it has the minimum increase in delay in relation to the previous node, the node is chosen to be search from in the next iteration (Step 1). This speeds up the algorithm considerably (and assures that there always is a (suboptimal) decision available, even if the algorithm is not ready yet).

The controller described so far uses perfect information. Now, however, in the sensitivity analysis, the input quantities are structurally disturbed, and the mathematical programming problem is solved considering these erroneous input quantities and their influence on the delay of the control system is evaluated. The different input quantities, i.e., predicted arrivals $a_i(k)$, predicted departures (saturation flow) $d_i(k)$, and current queue length $x_i(k_0)$, are disturbed one by one, leaving the others untouched, to see which input quantity is most sensitive to errors. For this chapter, it is assumed that the estimation or prediction method of the disturbed input quantity is biased (but no additional random noise is considered). For each input quantity a different independent structural error is introduced $\varepsilon_x, \varepsilon_a, \varepsilon_d \in (-1, \infty)$ for the queues $x_i(k_0)$, arrivals $a_i(k)$, and departures $d_i(k)$, respectively. The disturbed quantities are defined by:

$$\tilde{x}_i(k_0) = x_i(k_0)(1+\varepsilon_x) \quad \forall i \tag{2.7}$$
$$\tilde{a}_i(k) = a_i(k)(1+\varepsilon_a) \quad \forall i\ \forall k \tag{2.8}$$
$$\tilde{d}_i(k) = d_i(k)(1+\varepsilon_d) \quad \forall i\ \forall k \tag{2.9}$$

Note that the considered quantities cannot have negative values. Therefore $\varepsilon_x, \varepsilon_a, \varepsilon_d$ should be considered in the range $(-1, \infty)$. For errors $\varepsilon_x, \varepsilon_a, \varepsilon_d < 0$, the disturbed quantities become smaller than their original value, approaching 0 for errors approaching -1. For errors $\varepsilon_x, \varepsilon_a, \varepsilon_d > 0$, the disturbed quantities become larger than their original value. In this way, the error is varied in both directions, covering the whole possible range of values of the considered quantities. Further, note that the arrivals $a_i(k)$ and departures $d_i(k)$, are disturbed for the entire prediction horizon $\forall k \in [k_0, k_0+K]$. The queue information is only disturbed for the current state $k=k_0$, and the queue values for the remaining horizon follow from Equation 2.5. Finally, note that the structural errors are equal for each time interval and movement. Since the introduction of relative errors may result in non-integer values, the disturbed quantities are rounded downwards to the nearest integer, and the remaining part is transferred to the next time interval or movement, to assure on average the specified error percentages.

2.5.2 Experimental settings control scenario



	demand [veh/h/lane]		
	Vmax/C = 0.4	Vmax/C = 0.8	Vmax/C = 1.2
movement	undersaturated	saturated	oversaturated
1	130	270	450
2	140	280	480
3	30	70	110
4	30	50	90
5	290	570	970
6	30	50	90
7	140	280	470
8	140	280	470
9	110	210	360
10	30	50	90
11	150	290	500
12	30	50	90

*Vmax= sum of demands of maximum conflict group (2,5,9,12)

*C=capacity intersection (reduced by lost times) 1500 veh/h/lane

Figure 2.4: (a) Intersection configuration and demand scenarios, (b) cyclic control, (c) cyclic control with alternatives, (d) structure-free control.

The predictive control model is applied to a four-legged intersection with configuration as displayed in Figure 2.4a. The lanes are long enough, such that there is enough storage space for each direction and there is no spillback to the network entrances. Three different demand scenarios are chosen, representing the undersaturated (almost no queues present), saturated (queues present but mostly solved after green phase) and oversaturated case (queues remain after green phase). The saturated case will probably be the most interesting, since errors in the input quantities of the controller can result in insufficient green times, resulting in a collapsing system with high delays. The undersaturated and oversaturated cases are chosen for the purposes of comparison, to see if the control is indeed most sensitive in saturated cases. The demand scenarios are simulated for 30 minutes (time step 0.2 s). The arrivals are randomly distributed following an exponential arrival pattern with a constant mean (see Figure 2.4a for demand per movement). Each demand scenario is fixed to one repeatable realization.

Different types of predictive controllers are considered, varying in the degrees of freedom in the controller. The controllers all use (a subset of) the same fixed predefined movement groups but vary in the set of possible movement group transitions (Equation 2.3). The basic structures of the controllers are depicted in Figure 2.4b-d:

- Cyclic control with four movement groups. The main movement groups (1,2,5,6) are predefined and are only allowed in the cyclic predefined order. The cycle time differs per cycle resulting from the optimization process.
- Cyclic control with alternatives, i.e., four main movement groups and four additional movement groups. Next to the four main movement groups (1,2,5,6), more flexibility is added to the predefined cycle by considering four additional movement groups (3,4,7,8), that form an intermediate step between the main movement groups. The additional movement groups are optional, usage follows from the optimization process. The cycle time differs per cycle resulting from the optimization process.
- Structure-free control with all eight movement groups. Main movement groups and additional movement groups are considered equal. There is a free choice in the order of all the eight movement groups. The order of the movement groups follows from the optimization process. No cycles are imposed anymore (although they can arise from the optimization process).

For all control types, additional constraints are applied on lost times (all-red time of 3 s), minimum green times (3 s), maximum green times (30 s for through and left movements and 60 s for right movements).

As outlined in the predictive control model, the controllers are based on a rolling horizon approach. Signal states (and the active movement group) can change each time interval (6 s). Each new control interval (12 s), the control sequence is updated in real time considering a new planning horizon. The planning horizon is varied from 0 to 120 s. In this case five short cycles of 24 s of the main movement groups with a minimum duration of 6 s each (3 s lost time + 3 s minimum green) can be evaluated, or one long cycle of 120 s of the main movement groups is possible with maximum green durations of 30 s (3 s lost time included). This gives the cyclic controllers the possibility to adapt to fluctuations in the arrival pattern. Note that a very short planning horizon (0 s) coincides with non-predictive control, where the controller only reacts on already arrived vehicles.

As explained in Section 2.5.1, the arrivals are perfectly known beforehand, and can be fixed and stored in a preprocessing step for each demand scenario. The departures are approximated by experimentally derived saturation flow curves by measuring the queue discharge. The long-term saturation flow rate is one vehicle per 2 s. Using time intervals of 6 s, and an initial lost time of 3 s in the first interval, the discrete saturation flow curve is 1,2,2,3,3,3,3,… vehicles per time interval for increasing green duration. Using this preprocessed information, the controllers are all optimized on the fly, using the branch-and-bound solution method. The optimization problem is solved in real time, in 12 s, to come up with the new decision. In this time frame, exact solutions can be found for the cyclic controller and cyclic controller with alternatives for all planning horizons up to 120 s. For the structure-free controller exact solutions can be found up to planning horizon of 60 s, after that, the computation time becomes too long, and therefore suboptimal solutions are used.

2.5.3 Experimental results sensitivity analysis

Before the sensitivity analysis, as a reference, the performance of the control system is measured under perfect information. For the predictive controllers with increasing degrees of freedom (Figure 2.4), the performance of the system is analyzed for increasing prediction horizons. The results are presented in Figure 2.5 for the different demand scenarios. Note that the performance is expressed in delay per vehicle, obtained by dividing the total delay (objective) of the system by the number of vehicles, to get a more intuitive measure to compare the different demand scenarios. As Figure 2.5 shows, prediction improves the control system under perfect information. For all demand scenarios, the delay of the control system decreases for an increasing prediction horizon, and the performance is significantly better than for non-predictive control (considering a very short prediction horizon near zero).

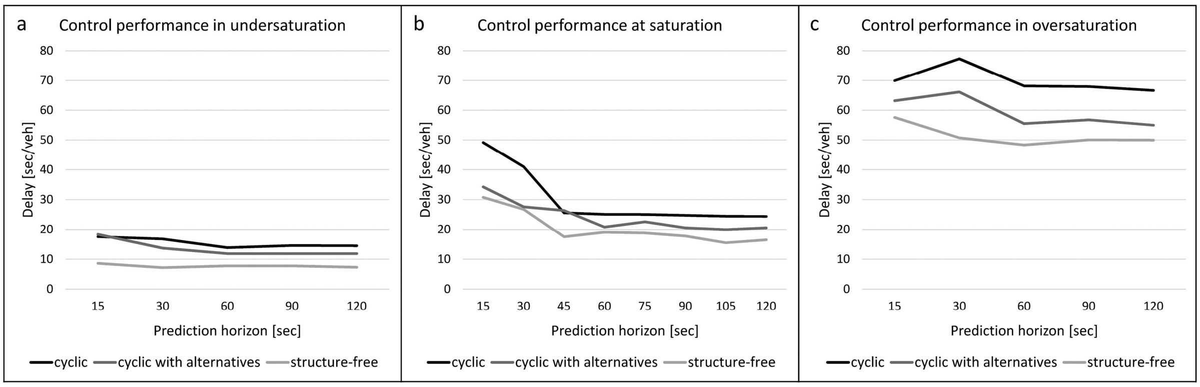


Figure 2.5: Performance of predictive controllers with perfect information on input quantities, for different demand scenarios: undersaturated (left), saturated (middle), oversaturated (right).

The structure-free controller with the highest degree of freedom, achieves better performance in terms of delay, and outperforms the cyclic controllers. There is a clear trade-off between adaptivity (degrees of freedom) and prediction horizon. The structure-free controller with high adaptivity can reach the same performance level with a short horizon, for which the cyclic controller with low adaptivity needs a much longer horizon. The gain of increasing the prediction horizon for the structure-free controller is less, than for the more constrained cyclic controllers.

In general, there is a gain in performance, by increasing the prediction horizon, although the gain in performance is less for longer horizons. For undersaturated conditions (Figure 2.5a), the line becomes flat from a certain prediction horizon. This point can be considered as the ideal prediction horizon length, i.e., the best performance can be obtained using this prediction horizon, and (almost) no additional performance can be gained when looking further ahead in the future. For saturated conditions, there is no monotone decreasing behavior (Figure 2.5b). Since finite prediction horizons are used, a suboptimal control optimum is reached if actions in the control horizon influence the traffic condition after the end of the prediction horizon (especially the case in highly saturated conditions). For different horizon lengths, the suboptimal solution is suboptimal in a different way, resulting in fluctuating performance levels. The ideal prediction horizon is less obvious in such situations and is approximated visually. For each demand scenario, the performance with perfect data is set as a reference (indexed to 100 for the structure-free controller).

Table 2.1: Overall results of sensitivity analysis.

quantity	error [%]	undersaturated						saturated						oversaturated					
		cyclic		cyclic with alternatives		structure-free		cyclic		cyclic with alternatives		structure-free		cyclic		cyclic with alternatives		structure-free	
		delay [idx]*	horizon [sec]	delay [idx]*	horizon [sec]	delay [idx]*	horizon [sec]	delay [idx]*	horizon [sec]	delay [idx]*	horizon [sec]	delay [idx]*	horizon [sec]	delay [idx]*	horizon [sec]	delay [idx]*	horizon [sec]	delay [idx]*	horizon [sec]
queue (state)	-100	288	60	198	60	121	30	193	90	163	90	159	90	168	90	160	90	144	90
	-50	225	60	165	60	100	30	162	60	131	90	125	60	144	60	115	60	103	90
	-20	193	60	165	60	100	30	152	60	126	60	108	60	139	60	114	60	100	60
	-10	193	60	165	60	100	30	148	60	125	60	101	60	139	60	114	60	100	60
	0	**193**	60	**165**	60	**100**	30	**148**	60	**124**	60	**100**	60	**138**	60	**114**	60	**100**	60
	10	193	60	165	60	100	30	148	60	126	60	101	60	139	60	114	60	100	60
	20	198	60	166	60	101	30	150	60	128	60	104	60	140	60	114	60	101	60
	50	226	60	168	60	104	30	156	90	140	60	121	90	145	60	115	60	109	60
	100	236	90	197	60	127	60	199	90	159	90	138	90	153	60	122	60	111	60
arrival pattern	-100	258	0	259	0	309	0	195	90	186	90	191	60	154	60	127	60	117	60
	-50	255	30	208	0	152	0	153	90	146	90	119	60	138	60	114	60	100	60
	-20	204	60	167	60	123	30	148	60	131	60	104	60	138	60	114	60	100	60
	-10	196	60	165	60	104	30	148	60	128	60	101	60	138	60	114	60	100	60
	0	**193**	60	**165**	60	**100**	30	**148**	60	**124**	60	**100**	60	**138**	60	**114**	60	**100**	60
	10	216	60	165	60	100	30	152	60	126	60	104	60	138	60	115	60	101	60
	20	218	60	165	60	100	30	158	60	130	60	111	60	138	60	118	60	105	60
	50	221	60	168	60	100	30	183	90	140	90	131	90	141	90	130	90	115	90
	100	253	60	213	60	100	30	198	90	188	90	193	90	169	90	156	90	142	90
departure pattern	-100	>400	30	>400	30	>400	60	>500	120	>500	90	>500	60	>300	90	>300	90	>300	90
	-50	268	30	223	30	148	60	236	120	213	90	228	60	162	90	156	90	149	90
	-20	217	60	188	60	113	30	174	90	138	90	155	60	145	60	129	60	113	60
	-10	212	60	167	60	111	30	158	60	131	60	135	60	142	60	122	60	105	60
	0	**193**	60	**165**	60	**100**	30	**148**	60	**124**	60	**100**	60	**138**	60	**114**	60	**100**	60
	10	202	60	165	60	103	30	148	60	126	60	108	60	138	60	114	60	100	60
	20	203	60	167	60	103	30	152	60	128	60	114	60	138	60	114	60	100	60
	50	203	60	173	60	118	30	164	30	137	30	139	30	138	60	114	60	101	60
	100	300	60	264	60	150	30	226	30	161	30	166	30	138	60	114	60	101	60

*Relative control performance expressed in delay indexes for the ideal prediction horizon (structure-free control with perfect data is set as a reference in boldface).

In the sensitivity analysis, the different input quantities, i.e., predicted arrivals, predicted departures (saturation flow), and current queue state, are one by one structurally disturbed to see which input quantity of the controller is most sensitive to errors. For the disturbed quantity, the errors ε_x , ε_a , ε_d (Equations 2.7-2.9) are taken from the set {-1, -0.5, -0.2, -0.1, 0, 0.1, 0.2, 0.5, 1}, and the performance of the controller is measured. Additionally, the relation between the performance and the horizon length is analyzed for the different error levels, and the ideal prediction horizon is determined to see if prediction still improves the performance of the system. In Table 2.1, this ideal prediction horizon and the performance are compared with the situation with perfect information. This is done for the different type of controllers, to see which controller is most sensitive, and for the different demand scenarios, to see if the results differ in undersaturated, saturated, and oversaturated conditions.

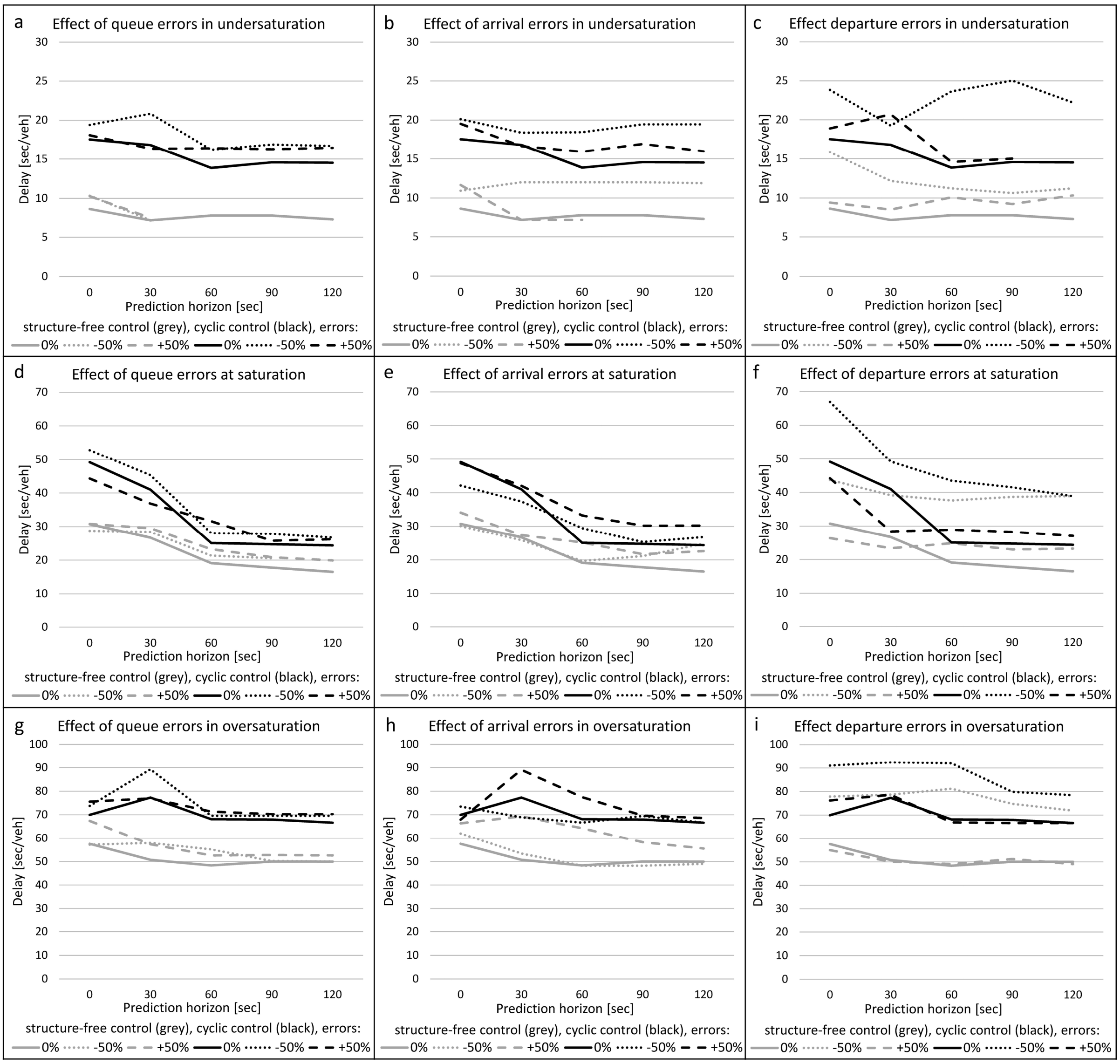


Figure 2.6: Performance of predictive controllers with errors in the input quantities: queues (left), arrivals (middle), departures (right), for different demand scenarios: undersaturated (top), saturated (middle), oversaturated (bottom).

As can be seen in Table 2.1, in general, increasing errors in the input quantities result in an increase in the total delay of the control system. In the end, an error in the input data results in too short or too long green times, yielding a drop in the performance of the control system. To be able to look in more detail into the behavior of the decrease of performance of the controllers, experimental relations between the delay and the prediction horizon for the different error levels are drawn. The results are presented in Figure 2.6, for the different control types and different demand scenarios. Only the sensitivity of the cyclic and structure-free controller is presented graphically, the results for the cyclic controller with alternatives can be found in Table 2.1. As expected, the results of the controller with alternatives lie in general between the results of the cyclic controller (fewer degrees of freedom) and structure-free controller (more degrees of freedom). Note that the experimental relations are presented for the 50% error levels, being extreme, however giving clear insight in the behavior of the system under errors. The behavior is similar for the lower error levels 10% or 20%, but less extreme, giving less additional delay (see Table 2.1). The following relations can be observed from the sensitivity analysis (answering the research questions at the beginning of this section):

Role of the prediction horizon

In most of the cases with erroneous input quantities (see Figure 2.6), prediction still leads to a better performance, i.e., the delay is decreasing for increasing prediction horizons. Increasing the prediction horizon can reduce the effect of errors. As can be seen in Figure 2.6a, in the undersaturated case, the effect of a disturbance in the queue of +50%, is eliminated completely for the structure-free controller (and partly for the cyclic controller). This can be explained by the fact that, in the undersaturated case, there are hardly any queues, so an error in the queue information has no large influence (except when completely ignoring the queues as in the -100% case in Table 2.1).

A reduction of errors for increasing prediction horizons is not guaranteed, however. As can be seen in Figure 2.6b, in which for the structure-free controller an underestimation in the arrival pattern of -50%, results in a remaining high delay for the larger prediction horizons. In this case, it is better to use no predicted arrivals at all, and purely react to arrived vehicles. Since in the undersaturated case there are hardly any queues, the arrivals are the most important information source. Therefore, errors in the arrival pattern will indeed influence the performance and these errors are accumulated for longer horizons.

Role of the type of controller

As can be noticed from Figure 2.6, in all demand scenarios and for all disturbed quantities, the structure-free controller is not more sensitive to errors in the input data than the cyclic controller. In most cases, the cyclic controller even seems to have a larger drop in performance. There seems to be a trade-off between sensitivity and adaptivity. The structure-free controller, with a high degree of freedom, can adapt better to fluctuations in the traffic conditions, than the more constrained cyclic controller. Although a structure-free controller relies more on the erroneous information, because of its adaptivity it can also react and correct mistakes more easily. Therefore, controllers with high adaptivity seem to outperform controllers with low adaptivity, even under erroneous predictions.

Role of the different input quantities

The control system is most sensitive for an error in the saturation flow (departure pattern), especially for an underestimation (-50% error) of the saturation flow, in saturated and oversaturated conditions when queues are present (Figure 2.6f). Note that in the undersaturated case with hardly any queues an error in the saturation flow can also have a large influence, building up queues quickly (Figure 2.6c). This sensitivity for the underestimation of the saturation flow, can already be noticed at the lower error levels of -10% and -20%, see Table 2.1.

The system is less sensitive for errors in the current queues and the predicted arrivals. This can be explained by compensating quantities, especially in the saturated case with moderate queues. If there are errors in the queue (Figure 2.6d), the system can rely on the perfect information on arrivals, and, the other way around, if there are errors in the arrivals (Figure 2.6e), the system can use the perfect information of the queues. In general, an error in a quantity does not need to be a problem, as long as other information can compensate for this error.

Role of objective of the control system

In Figure 2.6d-f, especially for the cyclic controller in the saturated case, some counter-intuitive effects can be noticed. For a +50% error in the queues (Figure 2.6d), or a -50% error in the arrivals (Figure 2.6e), for short prediction horizons, the performance is better than for the reference case of perfect information. This can be explained by the fact that, for short horizons, the total delay costs do not fully represent the real costs encountered when considering the entire horizon. The contribution of queues, especially, is not fully incorporated for short horizons. When there is an overestimation of the queues, or an underestimation of the arrivals, implicitly weights are given to the vehicles in the queue, resulting in more representative costs. This results in an overall decrease in the delay of the control system.

2.5.4 Design guidelines

From the results of the sensitivity analysis, it becomes clear that prediction indeed improves the performance of a controller if perfect information on predicted quantities is available. The delay of the system decreases for an increasing prediction horizon, and the performance is significantly better than for non-predictive control (considering a very short prediction horizon near zero). Therefore, adding a predictive component is of added value to the control system. However, in real life, the predictive information will never be perfect, and will contain errors. From the behavior of the control system under these errors, as studied in the sensitivity analysis, design guidelines can be defined. The following aspects need to be considered when designing predictive control:

Choice of prediction horizon

The choice of the prediction horizon strongly depends on the degrees of freedom of the controller. A more constrained controller asks for a longer prediction horizon to gain the full potential performance out of the controller.

The choice of the prediction horizon also depends on the level of saturation. If there are more queues present, control actions have a longer effect in time on the traffic conditions (large time delay of the dynamics in the control system). In this case a longer prediction horizon is needed to consider these effects in the optimization of the performance of the controller.

The choice of the prediction horizon also depends on the quality of the estimated and predicted input quantities of the controller. The use of a longer prediction horizon can reduce the effect of the input errors of the control system, resulting in a better performance. However, there is no guarantee that a longer horizon will automatically reduce the error and not amplify it (see previous section for examples of both effects). Therefore, in the design phase of a controller, using sensitivity analysis can give more insight into the behavior under errors of the controller for increasing prediction horizons, to choose the most convenient length of the prediction horizon.

Choice of type of predictive controller (degrees of freedom)

Under perfect information, a predictive controller with a high degree of freedom outperforms the more constrained controllers. There is a trade-off between adaptivity and prediction horizon. A controller with high adaptivity (a high degree of freedom) performs for a short horizon equally well as a controller with low adaptivity and a longer prediction horizon. In the design of the control system, the choice of a more adaptive controller with short prediction horizon or a more constrained controller with a longer prediction horizon can both increase the performance of the system.

Under disturbed conditions, although the controller with a high adaptivity relies more on the erroneous predictions, the controller is also more able to correct its mistakes more easily. There is a trade-off between sensitivity and adaptivity. In the design phase of a predictive controller, sensitivity analysis gives more insight into this interchanging behavior, to be able to choose the most convenient level of adaptivity of the controller.

Note that the choice of the type of predictive controller, will also depend on practical implementation issues. A high degree of freedom means a wider decision tree, resulting in longer computation times. A more constrained controller has a smaller decision tree, however, it needs a longer horizon, increasing the depth of the decision tree, which also finally results in longer computation times.

Choice and quality of input quantities

Less accurate predictions do not have to be a problem, if another quantity with enough accuracy is available that can be used to compensate the erroneous information. This makes the control system less sensitive for the predicted arrivals and the current queues, as these quantities contain compensating information. The control system is most sensitive for errors in the saturation flow that cannot be compensated by other information.

From the results of the sensitivity analysis, it can also be underlined that it is more important to predict that there is traffic waiting or arriving than how many vehicles are waiting or arriving (especially in undersaturated conditions). Predicting the arrival times, (and probably also the planned turning direction of the vehicles), is more important than an accurate prediction of the number of arrivals. This does not necessarily mean that simplified models that focus more on vehicle presence instead of vehicle numbers (e.g., simplified queue prediction models) achieve a similar control performance, however, to a lesser extent, the errors in vehicle numbers do influence the performance. To what extent a prediction model can be simplified before a substantial decrease in the performance occurs needs to be investigated in more detail.

Choice of objective of the control system

The results in this chapter are obtained for a predictive controller optimizing the delay of the traffic system. The choice of the control objective may influence the control decision and therefore the performance of the control system. However, most control strategies and objectives depend on the same quantities of the traffic system, like delay, number of stops, number of vehicles in the queue, and so on. Also, the traffic dynamics (queue formation) plays an important role in practically all formulations of performance of the control system, which is independent of the specific objective function. Therefore, mostly similar behaviour for these control systems is expected in the sensitivity on the considered input quantities in this chapter (predicted arrivals, predicted departures, measured queues).

In any case, it is important that the objective of the control system should represent the true costs of the control system for the entire horizon. If not, this can lead to counter-intuitive effects for shorter prediction horizons when input quantities are disturbed. Therefore, it may be helpful to include end costs in the objective function, reflecting the additional costs that vehicles will encounter beyond the limited prediction horizon (e.g., include end costs for queues that are still present at the end of the prediction horizon). It is left for further research to determine if this really limits the influence of information errors and leads to more robust controllers.

2.6 Conclusions and recommendations

In this chapter, an experimental framework was proposed to investigate the sensitivity of signalized traffic controllers for erroneous input quantities. The framework was illustrated for predictive control on a single intersection under different demand scenarios. Experimental relations between the performance of the control system and the prediction horizon were obtained for perfect information and erroneous input data. Different input quantities were structurally disturbed, concerning queue lengths (current state), number of predicted arrivals, and departures (saturation flow). These relations were studied for different types of predictive controllers with increasing levels of adaptivity (degrees of freedom), varying from cyclic to structure-free control.

The results show that prediction improves the performance of a signalized traffic controller, even in most of the cases with erroneous prediction information. Increasing the prediction horizon reduces the effect of errors and compensates errors with the information available from undisturbed predicted quantities (e.g., arrivals compensate queue information and vice versa). Therefore, controllers seem to be more sensitive to errors in stand-alone quantities (saturation flow in particular) that cannot be compensated by other information. Furthermore, controllers with a high adaptivity, and therefore a high ability to anticipate fluctuating traffic patterns, are not necessarily more sensitive to prediction errors. Although these controllers rely more on the erroneous information, controllers with high adaptivity can also react and correct mistakes more easily. Therefore, controllers with high adaptivity seem to outperform controllers with low adaptivity, even under erroneous predictions.

The outcome of the sensitivity analysis contributes to understanding the relations between information quality and performance of signalized traffic control. In the design phase of a controller, this insight can be used to make choices on the length of the prediction horizon, the level of adaptivity of the controller, the representativeness of the objective of the control system, and the input quantities that need to be estimated and predicted the most accurately.

The final goal of future research will be, on the one hand, to decide how accurate estimation and prediction methods should be to be of added value for signalized traffic control, and on the other hand, to be able to develop signalized traffic controllers that are robust to input errors. To this end, more experiments in this framework need to be done to analyze the effect of errors in the estimated and predicted input quantities on the performance of the controller. The experiments will be extended by disturbing quantities in different and more realistic ways as will be encountered in real life, like shifts in arrival times and directions, by introducing a random noise, and by combining errors of different quantities simultaneously. The experiments will be extended from a single intersection to coordinated intersections and finally a network context to represent real-life cases.

Chapter 3

Sensitivity analysis on the impact of prediction errors in network control

Model-based predictive signal control is a popular method to pro-actively control traffic and to reduce the effects of congestion in urban networks. In combination with structure-free controllers, which adapt signal settings in arbitrary order and combination (no imposed cycles), these predictive control methods have a high potential to increase system performance by adapting to individual vehicle patterns, which are increasingly available due to new technology. However, most of these control methods assume perfect predictions, while in practice there are prediction errors due to various reasons. In this chapter, the sensitivity of the system performance to these prediction errors is analyzed, for an urban corridor with spillback. In a microscopic simulator, first the ideal world is created for the structure-free model-based predictive signal controller, in which perfect predictions are made and the controller can reach its optimal performance. Then prediction errors are introduced in this perfect world, distinguished in aggregation errors that arise using a macroscopic prediction model and biases that represent structural errors in the prediction model or in its demand and state input. The effects of these prediction errors on the system performance are analyzed, as a function of the prediction horizon and update frequency of the control system. The results show that, even under errors, longer prediction horizons lead to better performance, up to a certain optimal prediction horizon length. A high update frequency dampens the influence of prediction errors, enabling the structure-free controller to correct mistakes faster. However, there remains a significant performance loss due to aggregation errors and biases in the prediction model, indicating a promising performance gain of more reliable predictions and the incorporation of information on individual vehicles in future control applications. Moreover, for all model quantities one direction of the bias has more impact on the system performance than the other direction, indicating guidelines towards a more robust control system that suffers less from erroneous predictions.

This chapter is based on the paper: M. C. Poelman, A. Hegyi, A. Verbraeck, and J. W. C. van Lint, "Structure-free model-based predictive signal control: A sensitivity analysis on a corridor with spillback," *Transp. Res. Part C Emerg. Technol.*, vol. 153, 104174, 2023. https://doi.org/10.1016/j.trc.2023.104174

3.1 Introduction

Traffic signal control is an important and widely used traffic management instrument to reduce the effects of congestion in urban networks, especially in saturated corridors where there is a high risk of spillback. Over the years, many signalized traffic controllers have been developed to guide traffic as efficiently as possible through the network (for an overview see [1],[11],[5]). Model-based predictive signal control is a popular approach, because by predicting the network wide traffic state and its evolution, the control decisions can be pro-actively optimized anticipating on future traffic conditions (for an overview see [19], [24]). However, most of these predictive signal control methods assume perfect predictions, while in practice there are prediction errors due to input errors, simplifications, and biases in the prediction models. In this chapter, the sensitivity to these prediction errors is investigated, analyzing the influence of prediction errors on the system performance of a model-based predictive controller in a corridor with spillback.

Model-based predictive signal control offers many possibilities to control the traffic system in an efficient manner [19], [24]. Since a prediction is made of the traffic state and its evolution, the controller makes pro-active decisions anticipating on future traffic conditions. The predictions are made by a traffic model, which makes it possible to calculate the effect of different possible control plans. An optimizer is used to select the control plan that is most effective according to the definition of performance in the specific control application. A model-based predictive controller uses a rolling horizon approach (as explained in [19], [24]). The control plan is optimized for the upcoming prediction horizon, after which the prediction horizon is shifted, and a new control decision is made. This assures a reinitialization of the controller to the current traffic state to correct for unforeseen situations and to update the prediction regularly. By increasing the prediction horizon of the controller, information on the states of different intersections is connected. Vehicles that are released at an intersection arrive at a downstream intersection later in time, and queues at an intersection propagate backwards causing spillback at upstream intersections. Looking ahead over multiple intersections allows to optimize the traffic state for a larger part of the network (network control), instead of optimizing each individual intersection separately (local control). Therefore, in theory, the performance of the control system increases by increasing the prediction horizon, especially in saturated networks with a high chance of spillback.

To reach an adequate performance level, the controller should provide enough control plans to choose from in the optimization. Most of the existing model-based predictive control methods use a pre-defined cyclic structure, optimizing cycle times and green splits only [19], [24]. The more advanced predictive control systems have a larger degree of freedom, i.e., a higher adaptivity level [11], adapt signal settings in arbitrary order and combination, and frequently (in seconds) reconsider signal settings dependent on arriving vehicles. Compared to the traditional pre-defined cyclic structure-based approaches, these structure-free predictive controllers essentially can match the fluctuations in the pattern of arriving vehicles better, improving the performance of the control system. However, these highly adaptive controllers require more computation time to find the optimal solution. Especially in the light of new technology, i.e., communicating vehicles and traffic lights, and new available data sources, i.e., online data on individual vehicles, highly adaptive controllers will be beneficial making full use of the new available data [5]. Therefore, structure-free (non-cyclic) model-based predictive signal controllers have a high potential controlling the traffic system in an efficient manner.

However, such control systems suffer from prediction errors when applied in real life. Often a macroscopic traffic flow model, for example the store-and-forward model [20] or a related model [36], is used for computational efficiency reasons to predict the evolution of the traffic state in the network [19], [24]. The individual driving behavior is aggregated in the prediction model, assuming equal average driving properties (like speed, acceleration, reaction times) for all vehicles. Individual destinations are aggregated as well into average turn fractions, and average demand patterns are assumed instead of individual vehicle arrivals, since these individual vehicle data are mostly not available yet. The aggregation process leads to a model mismatch between the macroscopic prediction model and how real traffic evolves, resulting in errors in the predicted traffic states. Moreover, also biases may be present in the control system [34], [29], [30]. The prediction model itself may contain biased assumptions on average driving behavior. The current traffic state, i.e., the initial state and basis of the prediction model, may be biased as well, since the traffic state, like queue length, cannot be directly measured, or is expensive to measure, and therefore needs to be estimated. The predicted arrivals and route choices, i.e., the demand input to the prediction model, can also be biased due to estimation errors. Biases in the input, initial state, and prediction model itself, lead to additional errors in the predicted traffic states. The prediction errors may propagate in the prediction model through the network, eventually leading to suboptimal control decisions and a decrease in system performance. Especially in a network with saturated traffic conditions and a high risk of spillback, small errors may have large consequences.

Existing work mainly focuses on improving data collection [6], [7] and estimation and prediction methods [9], [10]. With recent technological developments, data collection shifts from historical to real-time and from location-based to floating-car data [6], [7]. Advanced estimation and prediction methods are developed using these data to improve prediction accuracy and increase computational efficiency, where not only model-based methods but also data-driven approaches are used [9], [10]. However, until now, there has been very little study of the effect of data, estimation, and prediction errors on the performance of the signalized control system, as was already indicated for traffic management systems in general [28], and for model-based predictive signal control systems in particular [24], [29], [30]. Whereas, when designing and implementing these predictive signalized control systems, it is important to know which prediction errors cause the most performance decrease and which improvements in prediction accuracy will lead to the largest performance increase.

Therefore, this chapter studies the effect of prediction errors on the system performance of structure-free model-based predictive signal control by means of a sensitivity analysis. This study looks at the sensitivity to aggregation errors and additional biases in the prediction model, initial state, and demand input, that all sum up to prediction errors that may influence system performance. This chapter focuses on the sensitivity of structure-free model predictive control in a small network, i.e., a corridor with saturated conditions, where a small prediction error may have large consequences causing spillback. This study builds on the study in Chapter 2, where only a single intersection with local predictive control was analyzed. In this chapter, the sensitivity analysis is extended to a corridor of multiple intersections, considering spillback effects. The dependencies between up- and downstream intersections are considered in the sensitivity analysis, analyzing the effect of propagating prediction errors in the network on the total system performance. Moreover, this chapter studies the performance benefit of global predictive control with a total network objective and analyzes if this performance benefit is preserved under erroneous predictions. In the sensitivity analysis of the local controlled single intersection in Chapter 2 only biases were considered, i.e., structural errors in the initial state (queues), input (arrivals) and output (departures) of the prediction model. However, in a network context, macroscopic prediction models are often applied to simplify the traffic propagation modeling for computational efficiency reasons. Therefore, in this chapter, the prediction errors are separated into aggregation errors and additional biases in the sensitivity analysis, to identify the performance gain of more reliable predictions on the one hand and the incorporation of disaggregate individual vehicle information on the other hand. Moreover, in this study, not only errors in the initial state (queues) and input (arrivals) and output (departures) of the prediction model are analyzed, but also errors in the model itself, i.e., errors in the model parameters (travel speed, turning directions of the vehicles, saturation rate, and queue propagation speed), are considered, analyzing the model mismatch of the predicted traffic propagation through the network. Note that in Chapter 2 a comparison is made for a single intersection between structure-free and the more traditional cyclic model-based control. Since the structure-free controller outperforms the cyclic controller by adapting to individual vehicle patterns, with perfect as well as erroneous predictions (as was the conclusion in Chapter 2), the focus in this chapter lies on the structure-free model-based predictive controller, but now applied in a network context.

The final aim of this chapter is to give guidelines for developing structure-free model predictive control systems in a network. Various aspects in the sensitivity analysis are discussed, such as which prediction horizon length is still useful for looking ahead over intersections but not suffering too much from the error propagation, which update frequency is necessary to reduce prediction errors, and which quantity is most important to predict accurately in a network. Guidelines are presented towards the design of a more robust control system for a network that suffers less from erroneous predictions.

3.2 Methods

3.2.1 Experimental setup: general approach sensitivity analysis

In this research, a testbed environment is set up to analyze the sensitivity to prediction errors for a model predictive control system. The testbed environment is outlined in Figure 3.1. A microscopic simulator is used to represent the real world. A model predictive controller is connected to the simulation platform. Based on the current traffic state and the traffic demand as defined for the simulation, the controller optimizes the settings of the traffic signals by predicting the future traffic states in the network. The controller contains the standard components of a model predictive controller [19], [24]. A model is used to predict the future traffic state in the network for each candidate control plan. A control optimizer is used to select the optimal control plan, that has the largest predicted system performance for the entire network. A rolling horizon approach is applied to assure a reinitialization to the current traffic state. The control plan is optimized for the upcoming prediction horizon, after which the prediction horizon is shifted, and a new control decision is made. The optimal control plan is sent to the microscopic simulator, where the effects of the control decision are simulated. Afterwards, the performance of the control system is analyzed based on the simulation results.

In contrast to real life, all demand and state inputs to the model predictive controller are known exactly from the simulator in the testbed environment. Moreover, the prediction model can be tuned to reproduce the known and neat traffic behavior in the microscopic simulator as close as possible. Therefore, in the testbed environment, the controller can make the most accurate predictions (only a small tuning error remains), leading to an optimal control decision with optimal system performance. This provides us a clear reference situation for the sensitivity analysis. Prediction errors that arise in real life can be systematically introduced in the testbed environment. These prediction errors will eventually affect the solution obtained by the controller, resulting in non-optimal decisions. The effect of these non-optimal decisions on the system performance can be exactly measured in the microscopic simulation world. In the sensitivity analysis the effect of these prediction errors is analyzed and compared to the reference situation. Note that a macroscopic prediction model is chosen, which is mostly applied in real-life control applications for performance reasons. This allows us to study the effects of model mismatch between the macroscopic prediction and microscopic world that would occur in real life.

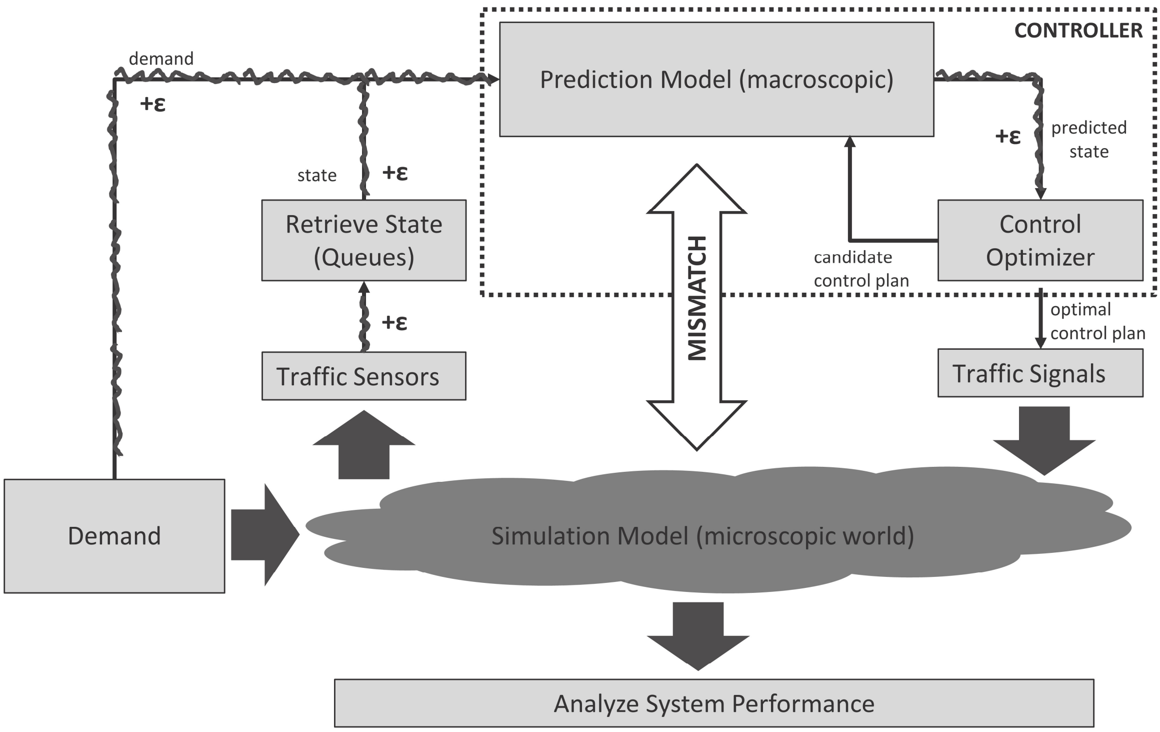


Figure 3.1: Model predictive control system in testbed environment. The wavy lines represent quantities that are affected by measurement, estimation, or prediction errors ε.

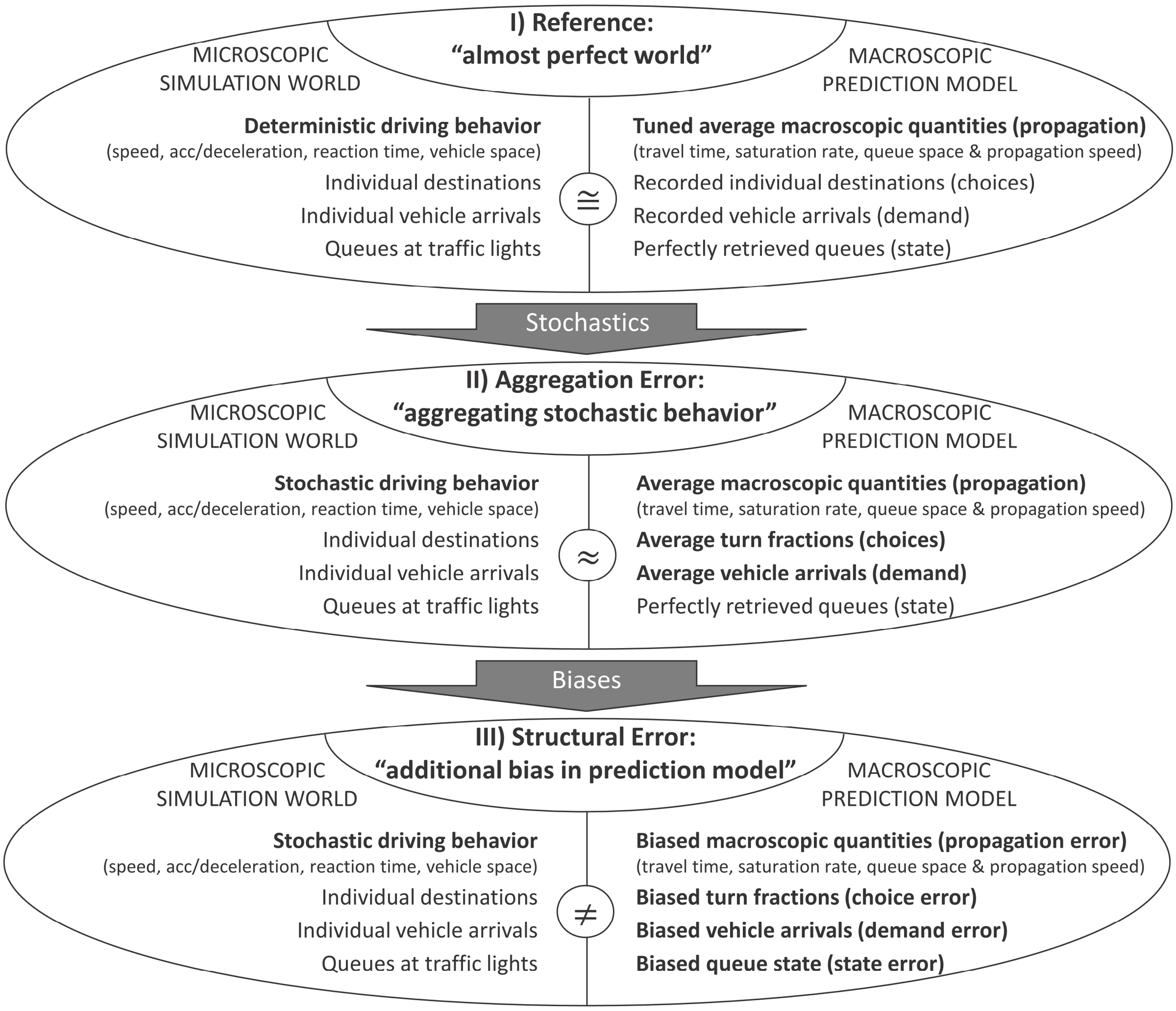


Figure 3.2: General approach of the sensitivity analysis.

The general approach of the sensitivity analysis is outlined in Figure 3.2. The sensitivity analysis consists of three steps. First, the almost perfect world is created to use as a reference situation (see step I in Figure 3.2). The almost perfect world represents the ideal world without prediction errors. In this almost perfect world, the predictions are made as accurate as possible. To this end, deterministic driving behavior is considered in the microscopic simulation, assuming all vehicles having equal driving parameters (like accelerations, decelerations, desired speeds, and reaction times). Drivers do not overtake and only change lanes at link exits if this is needed to reach their destination. The macroscopic prediction model is tuned to match the deterministic microscopic behavior, by measuring the values of the macroscopic parameters (like travel times, saturation rates, and queue propagation speeds) in the microscopic simulation. Moreover, individual vehicle information on arrivals and destinations is retrieved from the microscopic model and is assumed to be known in the macroscopic model as the specific demand pattern (including the turning directions of the vehicles). Also, the current queue state is retrieved from the microscopic simulator and assumed to be known exactly as a starting point for the macroscopic prediction. In this way, the macroscopic model can reproduce the microscopic world almost perfectly with its predictions (only a small tuning error remains), and the model predictive controller can find the optimal control decisions leading to the optimal system performance.

Second, the almost perfect world is disturbed by introducing stochastics (see step II in Figure 3.2). The deterministic driving behavior in the microscopic simulation is replaced by stochastic behavior. Vehicles are assumed to have different driving parameters, all varying around the original average values. Note that only longitudinal driving behavior is considered, lane change behavior due to overtaking is not considered in this study. The tuned macroscopic model no longer represents the exact behavior of all vehicles, but now corresponds to the average driving behavior. Moreover, the individual arrivals and destinations of the vehicles are assumed to be unknown in the macroscopic model and are replaced by the average demand pattern (including average turn fractions to indicate the distribution over the different directions). In fact, the stochastic behavior of the individual vehicles is aggregated in the macroscopic prediction model, resulting in aggregation errors. This represents the aggregation process in real life, that is inevitable when using a macroscopic model to predict microscopic behavior. The effect of these aggregation errors on the performance of the model predictive control system is analyzed.

Third, the aggregated world is disturbed further by introducing biases, i.e., structural asymmetric errors (see step III in Figure 3.2). These biases are introduced in the different components of the model predictive control system (as indicated by errors ε in Figure 3.1). The macroscopic prediction model itself is biased, by adding biases to the macroscopic model parameters (like travel times, saturation rates, and queue propagation speeds), representing a mismatch in driving behavior between the macroscopic prediction model and the microscopic world. The average demand input (including turn fractions) to the macroscopic prediction model is biased as well, representing demand prediction errors (and lack of information on vehicle destinations). Moreover, the initial queue state is biased, representing a combination of detection and state estimation errors that appear in real life. The effect of these biases on the system performance is analyzed.

Note that the approach of the sensitivity analysis is top down, disturbances are introduced in the almost perfect world and the (negative) effects of these prediction errors on the performance of the control system are analyzed. Bottom up the flow chart can be considered as the improvement of the control system, from the current imperfect control system based on biased macroscopic predictions, to an ‘ideal’ control system with perfect predictions including information on individual vehicles. Instead of analyzing the loss in performance by prediction errors, the gain in performance can be indicated of these methodological and technological improvements of the control system.

In the sensitivity analysis in this chapter, the focus lies on the network effects of prediction errors. Prediction errors propagate through the network in downstream as well as in upstream directions. Errors in predicted departures at an intersection will result in errors in the predicted arrivals at downstream intersections later in time. Errors in predicted queue states will affect the predicted departures at upstream intersections due to spillback. The longer the prediction horizon, the more the predicted state may deviate from the actual state to a point beyond which prediction may not be beneficial anymore. Therefore, the role of the prediction horizon will be considered in the sensitivity analysis. The prediction horizon is increased to look over multiple intersections, as well in downstream direction (travel direction of the vehicles) as in upstream direction (spillback direction), and the effect on the system performance is analyzed for perfect as well as erroneous conditions. For perfect predictions, it is analyzed how far the controller should look ahead to substantially increase performance, as there may be a horizon beyond which system performance does not improve anymore. For erroneous predictions, it is analyzed when the performance loss due to the error propagation overrides the benefit of the prediction. Note that updating the prediction and control decision more frequently based on (sufficiently accurate) actual queue states will probably reduce the effect of the error propagation. Therefore, the role of the update frequency will be considered in the sensitivity analysis as well. More specifically, the following aspects will be assessed:

- Relation between the prediction horizon and the performance of a control system with perfect predictions, to analyze the performance gain of an increasing prediction horizon in a network. And, to find the horizon (if any) beyond which system performance does not improve anymore.
- Relation between the prediction horizon and the performance of a control system with aggregation errors in the predictions, to analyze to what extent an increasing prediction horizon still increases system performance in a network. In other words, to analyze whether a macroscopic model is accurate enough to benefit from the prediction in the controller.
- Relation between the prediction horizon and the performance of a control system with biases in the predictions, to analyze to what extent an increasing prediction horizon increases the system performance in a network. Or, if not, to analyze to what extent the performance loss due to the error propagation in the network overrides the benefit of the prediction.
- Ranking of quantities to which the control system is most sensitive if not predicted correctly. In other words, ranking of quantities in order of importance to make accurate predictions in a network.
- Relation between the update frequency and the performance of a control system with predictions errors (aggregation errors or biases), to analyze to what extent a higher update frequency has a reducing effect on the performance loss due to error propagation in the network.

3.2.2 Technical setup: model predictive control system

Modeling framework

The modeling framework for the sensitivity analysis is an extension of the framework proposed in Chapter 2 for a single intersection, now being applied to a small network. A microscopic simulation model in Aimsun [37] is used to represent the real world. A model predictive controller is implemented and connected to the Aimsun simulation model. The controller is assumed to be structure-free, i.e., no cycles are imposed and the order in which the traffic signals become green together with the green duration is determined in the optimization process. In Chapter 2, the sensitivity analysis has shown that for a single intersection structure-free control outperforms model-based cyclic control even under erroneous conditions due to its high adaptive ability to correct bad decisions quickly. The prediction model in the structure-free controller is an extension of the macroscopic store-and-forward model of Chapter 2 including simple queue dynamics to predict spillback effects between the different intersections in the network. Note that the macroscopic network model is defined in enough detail containing the essential quantities to describe the microscopic traffic propagation but simplified enough to maintain a reasonable computational performance. The essential macroscopic model quantities are tuned to reproduce the reference and detuned in the sensitivity analysis. Moreover, the predictive controller is a network controller, i.e., it optimizes the future traffic situation in the entire network, making a control decision for all intersections simultaneously. Since a rolling horizon approach is applied, a series of subsequent optimization problems is solved. In each optimization problem, the control optimizer determines the optimal control sequence for the prediction horizon that minimizes the total delay in the network based on the predicted traffic states. Each optimization problem can be defined as a discrete mathematical programming problem. Note that the problem complexity of the structure-free model-based predictive controller rapidly grows with additional intersections, therefore, the exact solution method of Chapter 2 is no longer sufficient, and a heuristic solution method is needed and proposed in this chapter.

Mathematical formulation

To this end, the continuous time horizon is split into discrete control intervals $[(k\text{-}1)T, kT)$, with k the discrete time index referring to the k^{th} interval and T [s] the duration of the time interval. At the start of each control interval the signal setting at the intersections may change. Let i denote the index of a movement at any of the intersections in the network. Let m denote the index of an intersection. Predefine, per single intersection, a movement group with index j_m as a group of movements belonging to intersection m that are non-conflicting and are given green at the same time. Let $I(j_m)$ be the set of movement indices belonging to movement group j_m. Introduce signal states $s_i(k) \in \{0(\text{red}), 1(\text{green})\}$ and movement group states $p_{j_m}(k) \in \{0(\text{red}), 1(\text{green})\}$ for all movements i, movement groups j_m at intersections m, and time indices k. Exactly one movement group can be active at each intersection m at each time index k, i.e., $\sum_{j_m} p_{j_m}(k) = 1$. The states of the signals $i \in I(j_m)$, belonging to movement group j_m at intersection m, follow the state of this movement group at each time index k, i.e., $s_i(k) = p_{j_m}(k)$. There is no all-red movement group defined. Individual movements that are switching from state red to green, are assumed to stay red a loss-time T_L longer in the next control interval (assuming $T_L < T$) representing clearance times. There is a free choice of the active movement group at an intersection, there is no pre-defined order and there are no additional constraints concerning minimum or maximum green times. The objective of the controller is to find a control sequence of movement group states $p_{j_m}(k) \in \{0,1\}\ \forall j_m \forall m$, and corresponding signal states $s_i(k) \in \{0,1\}\ \forall i$, for the prediction horizon $k = k_0+1, \ldots, k_0+K$ that minimize the total delay over this prediction horizon, with k_0 referring to the current time interval the traffic system is in and K the number of prediction intervals looked ahead.

To predict the delay corresponding to a control sequence, a macroscopic store-and-forward model is used. In this model, vertical queues are assumed at the downstream end of each dedicated link of a movement. Vehicles are assumed to travel with equal free-flow speed to the downstream end of the link and enter the queue. Vehicles leave the queue when the signal is green, and if there is enough space at the dedicated links of the downstream movements. To determine the available space downstream the queues are assumed to be horizontal, and a simple queueing model is applied to determine the occupied space (head and tail) of the queue. According to these assumptions, a vehicle is queued and delayed if it arrived at the downstream end of the link according to free-flow conditions but has not departed yet due to a red signal, or queued earlier arrived vehicles, or a blockage downstream. The number of queued (and delayed) vehicles $x_i(k)$ [veh] per movement i is defined as:

$$x_i(k) \quad = x_i(k-1) + a_i(k) - d_i(k) \quad \forall i \, \forall k \tag{3.1}$$

with arrivals $a_i(k)$ [veh] at the queue and departures $d_i(k)$ [veh] from the queue, and $x_i(k_0)$ [veh] initialized to the current system state.

The arrivals at the queues of the external movements at the entrance of the network are equal to the demand. The arrivals at the queues of the internal movements are related to the departures of the queues of the upstream movements by:

$$a_i(k) = \textstyle\sum_{\tilde{\imath}} (d_{\tilde{\imath}}(k - n_i) * \pi_{\tilde{\imath}i}) \quad \forall i \, \forall k \tag{3.2}$$

where $\pi_{\tilde{\imath}i} \in [0,1]$ is the turn fraction of the vehicles leaving at upstream movement $\tilde{\imath}$ heading for movement i, and n_i is the number of intervals looking back upstream according to the free-flow travel time, i.e.,

$$n_i = \lceil (L_i/v_i)/T \rceil \quad \forall i \tag{3.3}$$

with L_i [m] the length and v_i [m/s] the free-flow speed of the dedicated link of movement i.

The departures from the queue are first determined by calculating the desired departures, considering unlimited space downstream, and, if necessary, will be restricted afterwards. The desired departures $\bar{d}_i(k)$ [veh] from the queue of movement i, are approximated by an average saturation flow rate r_i [veh/s] multiplied by the time duration T (corrected with the loss time $T_L \leq T$ when switching from red to green), i.e.,

$$\bar{d}_i(k) = \begin{cases} 0 & \text{if } s_i(k) = 0 \\ \min\{ r_i * (T - T_L), x_i(k-1) + a_i(k) \} & \text{if } s_i(k) = 1 \text{ and } s_i(k-1) = 0 \\ \min\{ r_i * T, x_i(k-1) + a_i(k) \} & \text{if } s_i(k) = 1 \text{ and } s_i(k-1) = 1 \end{cases} \quad \forall i \, \forall k \tag{3.4}$$

These desired departures $\bar{d}_i(k)$ [veh] from movement i, are limited by a maximum outflow $d_i^{\text{MAX}}(k)$ due to limited space at the downstream movements $\hat{\imath}$, resulting in the real departures $d_i(k)$ [veh], i.e.,

$$d_i(k) = \min\{ \bar{d}_i(k), d_i^{\text{MAX}}(k) \} \quad \forall i \, \forall k \tag{3.5}$$

with

$$d_i^{\text{MAX}}(k) = \min_{\hat{\imath}} \left\{ \frac{(L_{\hat{\imath}} - L_{\hat{\imath}}^{Q}(k-1))/L_{\text{VEH}} - \sum_{n=1}^{n_{\hat{\imath}}} \sum_{\tilde{\imath}} (d_{\tilde{\imath}}(k-n) * \pi_{\tilde{\imath}\hat{\imath}})}{\pi_{i\hat{\imath}}} \right\} \quad \forall i \, \forall k \tag{3.6}$$

where $\pi_{i\hat{\imath}} \in [0,1]$ is the turn fraction of the vehicles leaving movement i, heading for downstream movement $\hat{\imath}$, and $(L_{\hat{\imath}} - L_{\hat{\imath}}^{Q}(k-1))/L_{\text{VEH}}$ the available space expressed in vehicles at the downstream link of movement $\hat{\imath}$, i.e., link length $L_{\hat{\imath}}$ [m] minus the space occupied by the queue $L_{\hat{\imath}}^{Q}(k-1)$ [m] divided by the space of a single queued vehicle L_{VEH} [m], and $\sum_{n=1}^{n_i} \sum_{\bar{\imath}} (d_{\bar{\imath}}(k-n) * \pi_{\bar{\imath}\hat{\imath}})$ the number of driving vehicles already present on downstream movement $\hat{\imath}$. Note that $d_i(k)$ and $a_i(k)$ only depend on values of previous time indices, so the order in which the links of the different movements i are updated has no influence on the calculation.

The space occupied by the queue $L_i^Q(k)$ [m] in (3.6) is defined by the tail position of the queue. A simple (first order) queuing model is used to keep track of the tail and head position of the queue. The tail $L_i^Q(k)$ and head $L_i^H(k)$ of the queue are expressed in the positive number of meters in upstream direction measured from the downstream end of the link. The tail of the queue moves upstream with the arrivals, and remains active until the head of the queue, propagating upstream with constant speed v_i^H [m/s] $< v_i$ when releasing vehicles, meets the tail position, i.e.,

$$L_i^Q(k) = \begin{cases} L_i^Q(k-1) + a_i(k) * L_{\text{VEH}} & \text{if } L_i^H(k-1) < L_i^Q(k-1) \text{ and } (s_i(k) = 0 \text{ or } s_i(k-1) = 1) \\ x_i(k) * L_{\text{VEH}} & \text{otherwise ((re)initialization)} \end{cases} \quad \forall i \, \forall k \qquad (3.7)$$

and

$$L_i^H(k) = \begin{cases} L_i^H(k-1) + v_i^H * T & \text{if } 0 < L_i^H(k-1) < L_i^Q(k-1) \text{ and } (s_i(k) = 0 \text{ or } s_i(k-1) = 1) \\ v_i^H * (T - T_L) * 1_{[d_i(k)>0]} * 1_{[x_i(k)>0]} & \text{otherwise ((re)initialization)} \end{cases} \quad \forall i \, \forall k \qquad (3.8)$$

with $1_{[d_i(k)>0]}$ and $1_{[x_i(k)>0]}$ indicator functions, 1 if the condition holds, 0 otherwise. In the model a standing queue at the stop line is assumed the moment the signal turns green, i.e., $s_i(k) = 1$ and $s_i(k-1) = 0$. The tail $L_i^Q(k)$ of the queue is initialized to the (occupied space of the) total number of vehicles in this queue (initialization in (3.7)). From there the tail moves upstream with the arrivals. The head $L_i^H(k)$ of the queue is initialized to zero and starts to move when releasing vehicles at green, i.e., if $d_i(k) > 0$ and $x_i(k) > 0$ (initialization in (3.8)). The head continuous to move upstream with constant speed v_i^H while the head is smaller than the tail, i.e., $L_i^H(k-1) < L_i^Q(k-1)$. If the head meets the tail, the head and tail are reinitialized. If the queue is dissolved during green ($x_i(k) = 0$), then the head and tail will both be reset to zero. If there is a remaining queue ($x_i(k) > 0$) since the traffic signal turns red before the head meets the tail, the head will be reset to zero ($d_i(k) = 0$) and the tail will be reset to the (occupied space of the) total number of vehicles left in the queue. Note that the first order model is only accurate if the head meets the tail before the signal gets green again for a second time. Note that (3.7) and (3.8) force a reinitialization if this is not the case. In this chapter only short distances between intersections will be considered, resulting in short queues for which this queueing model is accurate enough.

According to the complete store-and-forward model, the optimization of delay of the predictive control system can be expressed as:

$$\min_{\{p_{j_m}(k), \forall j_m \forall m \forall k\}} \sum_i \sum_{k=k_0+1}^{k_0+K} x_i(k) * T \tag{3.9}$$

s.t.

$s_i(k) = p_{j_m}(k) \qquad \forall i \in I(j_m)\ \forall j_m\ \forall m\ \forall k$

$\sum_{j_m} p_{j_m}(k) = 1 \qquad \forall m\ \forall k$

$x_i(k)$ given by (3.1)-(3.8) $\qquad \forall i\ \forall k$

Solution method

The discrete mathematical programming problem (3.9) with the explicit formulation of the store-and-forward model (3.1)-(3.8) can be solved by searching through a corresponding decision tree. The starting node of the tree is the current state of the control system $\{x_i(k_0)\ \forall i\}$, branches are formed by the possible decisions for the next time interval k_0+1, i.e., for all intersections m the choice of the active movement group j_m for which $p_{j_m}(k_0+1)=1$, resulting in nodes at the next time layer with state $\{x_i(k_0+1)\ \forall i\ \}$, and so on, adding time layers for $k = k_0+1, \ldots, k_0+K$. The optimal decision sequence is the path in the tree with minimum delay. The problem size, given by the number of possible paths $(|J|\text{^}|M|)\text{^}K$, rapidly grows with the width of the tree, determined by the combination $(|J|\text{^}|M|)$ of the average number of movement groups $|J|$ per intersection and the number of intersections $|M|$, together with the depth of the tree, given by the number of prediction intervals K. Since the problem size can already be large in a small network, heuristic approaches are needed in the search process, to find a near optimal solution in acceptable time.

The applied heuristic solution approach is outlined in Figure 3.3b. The rolling prediction horizon $k = k_0+1, \ldots, k_0+K$ is split into a control horizon $k = k_0+1, \ldots, k_0+K^{CTR}$, $K^{CTR} \leq K$, where the signal states are fully optimized and (if $K^{CTR} < K$) a tail $k = k_0+ K^{CTR}+1, \ldots, k_0+K$, where the search space is limited, and suboptimal signal states are obtained (Figure 3.3b). Note that in the rolling horizon approach, the solution is updated after a pre-defined period of K^{UP} intervals. The control horizon $k = k_0+1, \ldots, k_0+ K^{CTR}$ is chosen at least as large as this update horizon $k = k_0+1, \ldots, k_0+ K^{UP}$, $K^{UP} \leq K^{CTR} \leq K$. In this way, only the fully optimized signal states are implemented, and the suboptimal tail is reconsidered and optimized further in the next solution update. In this chapter, for performance reasons K^{CTR} is chosen as small as possible, i.e., $K^{CTR} = K^{UP}$, although not smaller than 2 intervals to assure enough degrees of freedom in the decision sequence.

The optimal solution in the tail of the decision sequence is approximated by the greedy solution, i.e., in each time interval the control decision is made that provides the most performance gain in this time interval, without considering the effects of this decision for the remaining horizon. The greedy solution can be calculated efficiently but is typically a suboptimal solution. The head of the decision sequence for the control horizon is fully optimized by a branch-and-bound approach. This full optimization is more costly, but an exact solution can be obtained. Note that the greedy tail itself is not part of the implemented solution and will be used as a starting point to optimize further in the next decision update. The greedy tail is determined by solving subsequent optimization problems, i.e., $\min_{\{p_{j_m}(k), \forall j_m \forall m\}} \sum_i x_i(k) * T$ given state $\{x_i(k\text{-}1)\ \forall i\ \}$ sequentially for $k = k_0+ K^{CTR}+1, \ldots, k_0+K$, and is in fact constructed by following the greedy branch from a given node in layer k-1 to the next layer k in the tree.

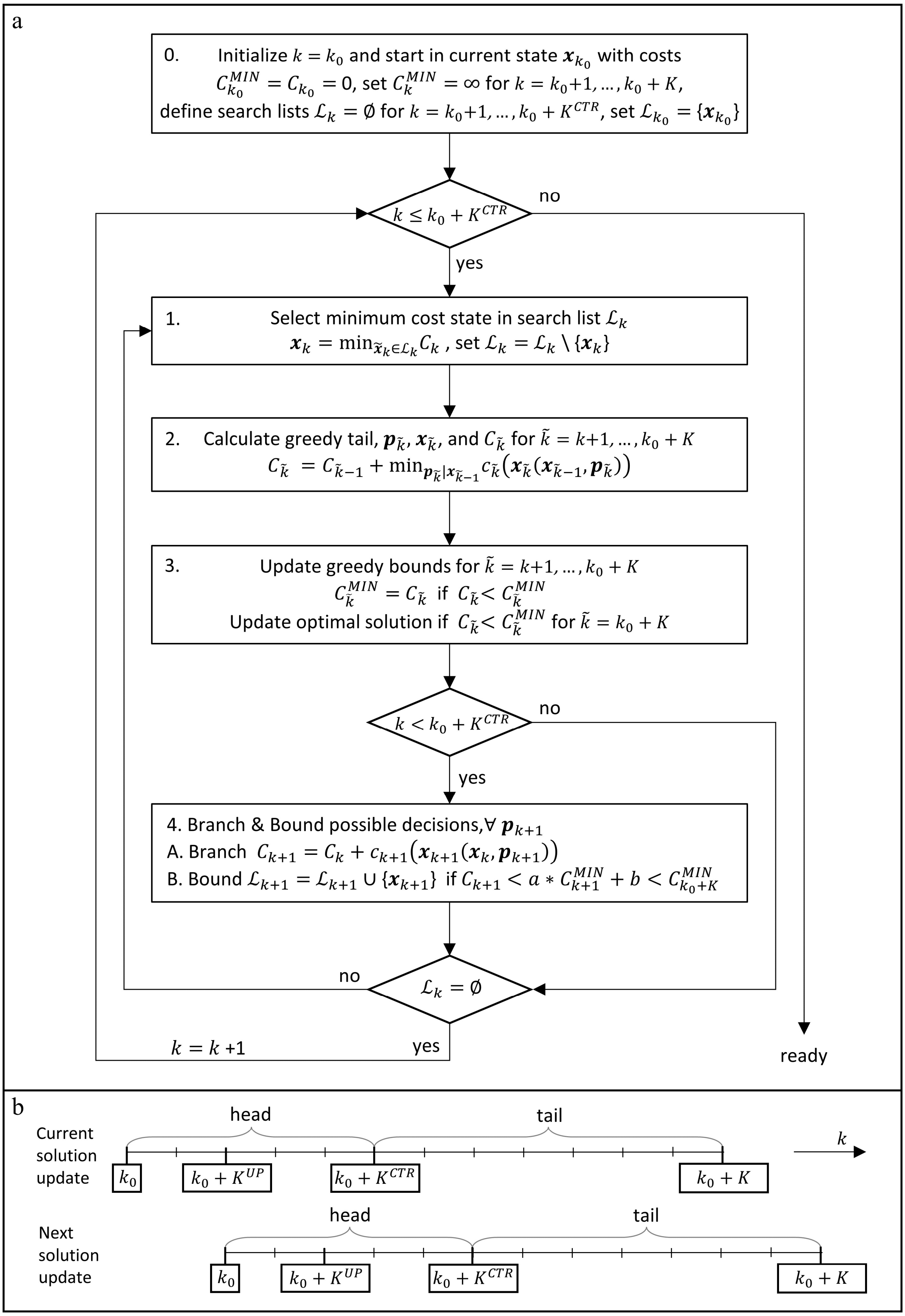


Figure 3.3 (a,b): Flowchart of the branch-and-bound heuristic with greedy tail.

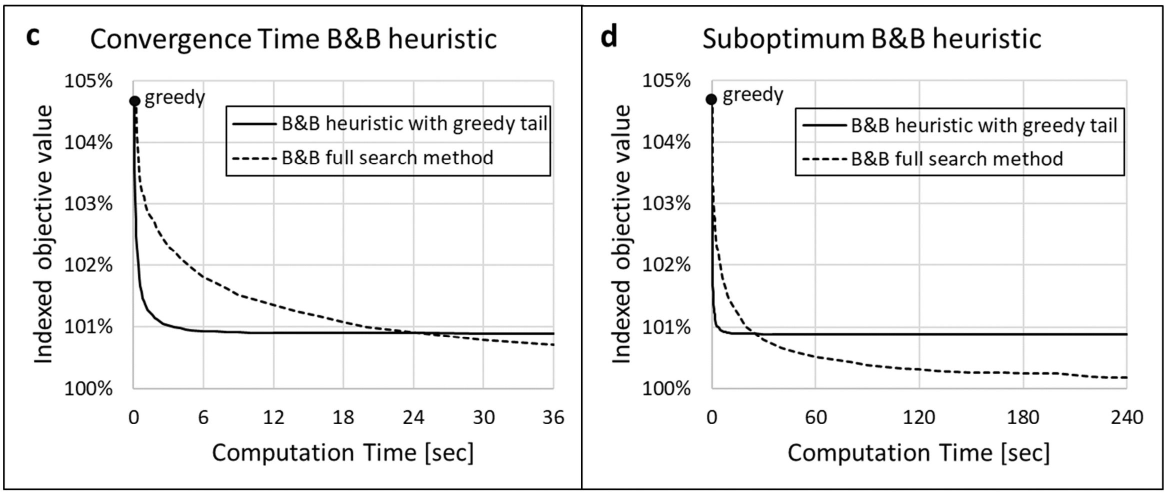


Figure 3.3 (c,d): Convergence of the branch-and-bound heuristic with greedy tail indexed to the optimal solution level of the full branch-and-bound search method, timed on a computer with an Intel i7-6820HQ processor.

Mathematically, the objective of the controller in (3.9), is approximated by:

$$\min_{\{\boldsymbol{p}_k\}_{k=k_0+1}^{k_0+K^{CTR}}} \left(\sum_{k=k_0+1}^{k_0+K^{CTR}} c_k\big(\boldsymbol{x}_k(\boldsymbol{x}_{k-1}, \boldsymbol{p}_k)\big) + \sum_{\tilde{k}=k_0+K^{CTR}+1}^{k_0+K} \min_{\boldsymbol{p}_{\tilde{k}}|\boldsymbol{x}_{\tilde{k}-1}} c_{\tilde{k}}\big(\boldsymbol{x}_{\tilde{k}}(\boldsymbol{x}_{\tilde{k}-1}, \boldsymbol{p}_{\tilde{k}})\big) \right) \quad (3.10)$$

with $\boldsymbol{x}_k$ the vector of queue states $x_i(k)\ \forall i$, $\boldsymbol{p}_k$ the vector of movement group states $p_{j_m}(k), \forall j_m\ \forall m$, and c_k the cost for time index k as a function of the queue state vector $\boldsymbol{x}_k$, i.e., $c_k(\boldsymbol{x}_k) = \sum_i x_i(k) * T$. Note that $\boldsymbol{x}_k$ depends on $\boldsymbol{x}_{k-1}$ and $\boldsymbol{p}_k$ through (3.1)-(3.8), that are defined in explicit form. Note that the costs are additive over time, and can be updated cumulatively, $C_k = C_{k-1} + c_k$, with C_k the cumulative costs for $k = k_0+1, \ldots, k_0+K$ and $C_{k_0} = 0$. Therefore a branch-and-bound tree search can be followed to find the solution to (3.10). Note that the greedy solution can be used as an initial solution for the complete horizon, setting the initial cost bound in the search process, i.e., $C_{k_0+K} = \sum_{\tilde{k}=k_0+1}^{k_0+K} \min_{\boldsymbol{p}_{\tilde{k}}|\boldsymbol{x}_{\tilde{k}-1}} c_{\tilde{k}}\big(\boldsymbol{x}_{\tilde{k}}(\boldsymbol{x}_{\tilde{k}-1}, \boldsymbol{p}_{\tilde{k}})\big)$.

The branch-and-bound algorithm with greedy tail is presented in Figure 3.3a. The search process starts at the first layer of the tree, i.e., at time index k_0 (Step 0). For the active time layer k, the node with the lowest costs C_k is selected. Note that for time layer k_0 the start node of the tree is selected, i.e., current state $\boldsymbol{x}_{k_0}$ with $C_{k_0} = 0$ (Step 1). First the greedy path from the active node to the end of the tree is determined (Step 2). The costs along the greedy path, $C_{\tilde{k}}$ for $\tilde{k} = k+1, \ldots, k_0+K$ are used to determine the minimum cumulative costs $C_{\tilde{k}}^{\text{MIN}}$ found so far at each layer. These optimal layer costs are used later to limit the search space. Note that for time layer $\tilde{k} = k_0+K$ the minimum costs $C_{\tilde{k}}^{\text{MIN}}$ corresponds to the best complete solution found so far, that is updated if $C_{\tilde{k}} < C_{\tilde{k}}^{\text{MIN}}$ (Step 3). Then the active node is branched to the next time layer $k+1$, by considering all possible decisions $\boldsymbol{p}_{k+1}$, calculating new states $\boldsymbol{x}_{k+1}$ with costs c_{k+1} and updating cumulative costs $C_{k+1} = C_k + c_{k+1}$ (Step 4A). The branched nodes are bounded if the costs are larger than the complete solution costs found so far, i.e., $C_{k+1} \geq C_{k_0+K}^{\text{MIN}}$ or if the costs are too large compared to the minimum cumulative layer costs C_{k+1}^{MIN}, i.e., $C_{k+1} \geq a * C_{k+1}^{\text{MIN}} + b$ with a and b parameters to restrict the search space. These parameters are chosen large enough to include ineffective intermediate solutions that could still lead to the optimal solution, but not too large to find a solution in a reasonable amount of time (Step 4B). Accepted nodes are added to a search list. The process is repeated until all accepted nodes are searched through for the head of the decision sequence, i.e., for time layers $k \leq k_0 + K^{CTR}$. The final solution is the last updated optimal solution (result of step 3).

In Figure 3.3c and 3.3d the convergence of the branch-and-bound heuristic with greedy tail is compared to the full branch-and-bound search, where not only the head but also the tail is fully optimized. The average convergence is shown for the reference of the corridor testcase (Section 3.2.3), with problem complexity $|M| = 4$ intersections, $|J| = 4$ movement groups per intersection, and a prediction horizon of $K = 10$ intervals with length $T = 6$ seconds. Both methods start in the greedy solution and use the same search space (a, b values). The heuristic method optimizes only the first two intervals, $K^{CTR} = 2$, assumes a greedy tail and reconsiders part of the greedy tail in the next solution update, whereas the full search method optimizes all intervals at once ($K^{CTR} = K = 10$). The full search method converges steadily, reaching its optimal solution level after 240 seconds (see Figure 3.3d), however, takes too much time (slower than real time) if an update interval of K^{UP}=1 (6 seconds) is considered. The heuristic with greedy tail converges much faster (see Figure 3.3c), within 2 seconds (2/12 of the available update time if K^{UP} =2, and 2/6 if K^{UP} =1), reaching a suboptimal solution that lies close (<1%) to the optimal solution of the full search method (see Figure 3.3d). Therefore, the heuristic solution approach is efficient and accurate enough for the problem analyzed in this chapter.

Tuning and detuning

To create the “almost perfect world” in the first step of the sensitivity analysis, the macroscopic prediction model is tuned to be equal to the deterministic microscopic driving behavior (step I in Figure 3.2). This is done by measuring the values of the macroscopic parameters of model (3.1)-(3.8) in the microscopic simulation. The free-flow speed v_i [m/s] in (3.3) is tuned by measuring the travel time of the vehicles from link exit to link exit under free-flow conditions (and green signals). Note that an intersection has no space in the macroscopic model, and travel times on the intersection surface need to be incorporated in the (therefore slightly slower) macroscopic link speed. The saturation flow rate r_i [veh/s] in (3.4) is tuned by measuring the individual passage times at the link exit of vehicles leaving a queue from the moment the signal becomes green. The individual passages are averaged into the saturation flow rate r_i [veh/s]. Note that the tuned value of the saturation flow rate is slightly lower for the first intervals after turning green compared to the subsequent intervals, caused by the reaction times when switching signal states. The maximum storage capacity L_i/L_{VEH} at a link in (3.6) is tuned by measuring the distance headway L_{VEH} [m] in a standing queue. Note that the actual available space follows from the queueing model (3.7)-(3.8). The queue head propagation speed v_i^H [m/s] in (3.8) is tuned by measuring the times the vehicles start to move in the queue from the moment the signal turns green. The individual time headways are averaged and translated into a constant propagation speed (using the tuned distance headway L_{VEH} [m]). In addition to the tuning of the macroscopic model parameters, the demand input to the macroscopic prediction model is exactly known and is directly retrieved from the microscopic simulator. The individual demand pattern is recorded in advance for the entire simulation. The arrivals $a_i(k)$ [veh] in (3.1) at the external movements at the entrance links of the network are set equal to the (aggregated) recorded arrivals. Moreover, the turn fractions $\pi_{i\bar{i}} \in [0,1]$ in (3.2) and (3.6) are replaced by the recorded real destinations of the vehicles and become time dependent. The initial state input to the prediction model is exactly known as well and is derived from the simulation data. During the simulation, the individual vehicle passages at the link exits are measured and aggregated into the macroscopic departures $d_i(k)$, known up to time index k_0. Based on these simulation data, the initial macroscopic state of the queues $x_i(k_0)$ [veh] is calculated accordingly (3.1)-(3.3). With the perfect state initialization, exact demand pattern, and tuned model parameters, the macroscopic model can reproduce the microscopic world close enough to function as a reference in the sensitivity analysis (as will be shown in Section 3.3.1).

In the second step of the sensitivity analysis, the effect on the system performance of aggregation errors in the prediction model is analyzed (step II in Figure 3.2). To this end, the deterministic driving behavior in the microscopic simulator is replaced by stochastic behavior (with parameters around the original average values). So, the tuned macroscopic prediction model now corresponds to the aggregated average driving behavior. Moreover, the individual demand pattern is replaced by the average demand pattern as input for the macroscopic prediction model. The arrivals $a_i(k)$ [veh] in (3.1) at the external movements at the entrance links of the network are set back to the average demand values. The turn fractions $\pi_{i\hat{\imath}} \in [0,1]$ in (3.2) and (3.6) are set back to the average fractional values instead of considering real destinations of the vehicles. The effects on the system performance of all these aggregation errors are measured.

In the third step of the sensitivity analysis, the effect of biases in the different components of the macroscopic model predictive control system on the system performance are analyzed (step III in Figure 3.2). To this end, the tuned macroscopic model parameters, i.e., free-flow speed v_i [m/s] in (3.3), saturation flow rate r_i [veh/s] in (3.4), queue head propagation speed v_i^H [m/s] in (3.8), are biased by introducing structural errors, i.e., $\tilde{r}_i = (1 + e_r) * r_i\ \forall i$, $\tilde{v}_i = (1 + e_v) * v_i\ \forall i$, $\tilde{v}_i^H = (1 + e_{v^H}) * v_i^H\ \forall i$, with $e_r, e_v, e_{v^H} \in [-1,1]$. Additionally, the average demand pattern is biased by introducing a structural error in the average arrivals $a_i(k)$ [veh] in (3.1) at the network entrances, i.e., $\tilde{a}_i(k) = (1 + e_a) * a_i(k)\ \forall k \forall i$, with $e_a \in [-1,1]$. Biased turn fractions $\pi_{i\hat{\imath}} \in [0,1]$ in (3.2) and (3.6) are considered as well, including a structural error in the main direction from movement i to downstream movement $\hat{\imath}$, i.e., $\tilde{\pi}_{i\hat{\imath}} = (1 + e_\pi) * \pi_{i\hat{\imath}}$, with $e_\pi \in [-1, (1 - \pi_{i\hat{\imath}})/\pi_{i\hat{\imath}}]$, and correcting the other directions to downstream movements $\bar{\imath}$, i.e., $\tilde{\pi}_{i\bar{\imath}} = (1 - (\pi_{i\hat{\imath}}/(1 - \pi_{i\hat{\imath}}))\ * e_\pi) * \pi_{i\bar{\imath}}\ \ \forall \bar{\imath} \neq \hat{\imath}$, such that $\sum_{\bar{\imath}} \tilde{\pi}_{i\bar{\imath}} = 1$. Finally, the initial queue states $x_i(k_0)$ [veh] in (3.1) are biased by introducing structural errors, i.e., $\tilde{x}_i(k_0) = (1 + e_x) * x_i(k_0)\ \forall i$, with $e_x \in [-1,1]$. The effects on the system performance of these biases are measured.

Note that the performance of the control system in the microscopic world can always be measured exactly. To preserve comparability between both worlds, the microscopic system performance is expressed in macroscopic quantities, according to the macroscopic definition of queues and delay (Equations (3.1)-(3.3), with measured departures $d_i(k)$, recorded arrivals $a_i(k)$, and real destinations for the turn fractions $\pi_{\bar{\imath}i} \in [0,1]$).

3.2.3 Traffic scenario: corridor with spillback

Network configuration

The sensitivity analysis is performed on a corridor of four intersections, where queues can cause spillback and block traffic at upstream intersections (see Figure 3.4b). The intersections are connected by short low-speed links (< 50 km/h), representing urban settings. The distance between the intersections is small (smaller than in real life) to cause spillback in less simulation time and needing fewer vehicles, where a similar process will arise in real life covering more time and involving more vehicles. Each intersection consists of the full twelve movements and considers only cars. Each movement has a separate lane, where only vehicles for this direction are allowed. Lane changes only take place at the intersections and are not allowed in the links. Vehicles only cross the intersection if there is enough space downstream and if they do not block the intersection surface. These assumptions are a simplification of reality, in which different vehicle types are mixed and driving behavior is less neat causing more incidental and severe peaks of spillback. These simplifications were done to reduce the stochastics in the sensitivity analysis experiments to create a solid reference situation ("the almost perfect world") and to isolate and focus on the effects of prediction errors.

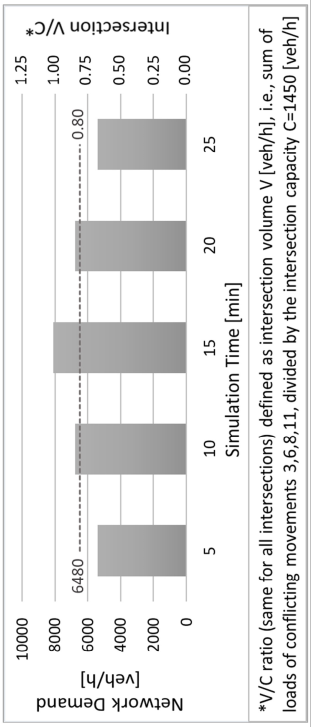
Controller

The corridor is controlled by a structure-free model predictive signal controller as defined by (3.9). At each intersection each movement has its own signal and can be controlled separately. Signal states may switch every 6 seconds (T=6), considering a loss time of 3 seconds (T_L=3). At each intersection there are 4 movement groups defined of non-conflicting movements that have green at the same time (see Figure 3.4c). The controller has a free choice among those 4 movement groups at each intersection, and there is no predefined order and no imposed cycle (and no other constraint on maximum waiting or green times). Moreover, the decision for a specific movement group at one intersection is independent from the choice at the other intersections, resulting in 4^4=256 possible decisions in each time interval of the prediction horizon (and 256^K possible decision sequences for the entire prediction horizon of K intervals). Note that in theory there are 8 possible movement groups per intersection (and even more when considering public transport, bikes, and pedestrians), resulting in an increasing amount of possible decision sequences. To reduce problem complexity, only the 4 most common movement groups are considered, without losing the principle of a structure-free controller.

To analyze the benefit of the prediction, the prediction horizon of the controller is increased from 6 to 120 seconds in the experiments (K=1,2, … ,20). Note that 2 minutes seems enough to capture most of the traffic dynamics for this size of corridor. The controller in principle updates the decision sequence after each 12 seconds (K^{UP}=2). However, a higher update frequency of each 6 seconds (K^{UP}=1) is considered as well to analyze the possible reduction of prediction errors, and lower update frequencies (K^{UP}=5 or 10) are analyzed for possible opposite effects. The role of the prediction horizon and the update frequency of the controller is analyzed in the sensitivity analysis under perfect as well as erroneous predictions.

a) demand

from\to [veh/h]	1	2	3	4	5	6	7	8	9	10	total
1		194	130	194	130					259	907
2	194		259				130				583
3	130	259				194					583
4	194				259				130		583
5	130			259				194			583
6			194				259			130	583
7		130				259				194	583
8					194				259	130	583
9				130				259		194	583
10	259					130	194	130	194		907
total	907	583	583	583	583	583	583	583	583	907	6480



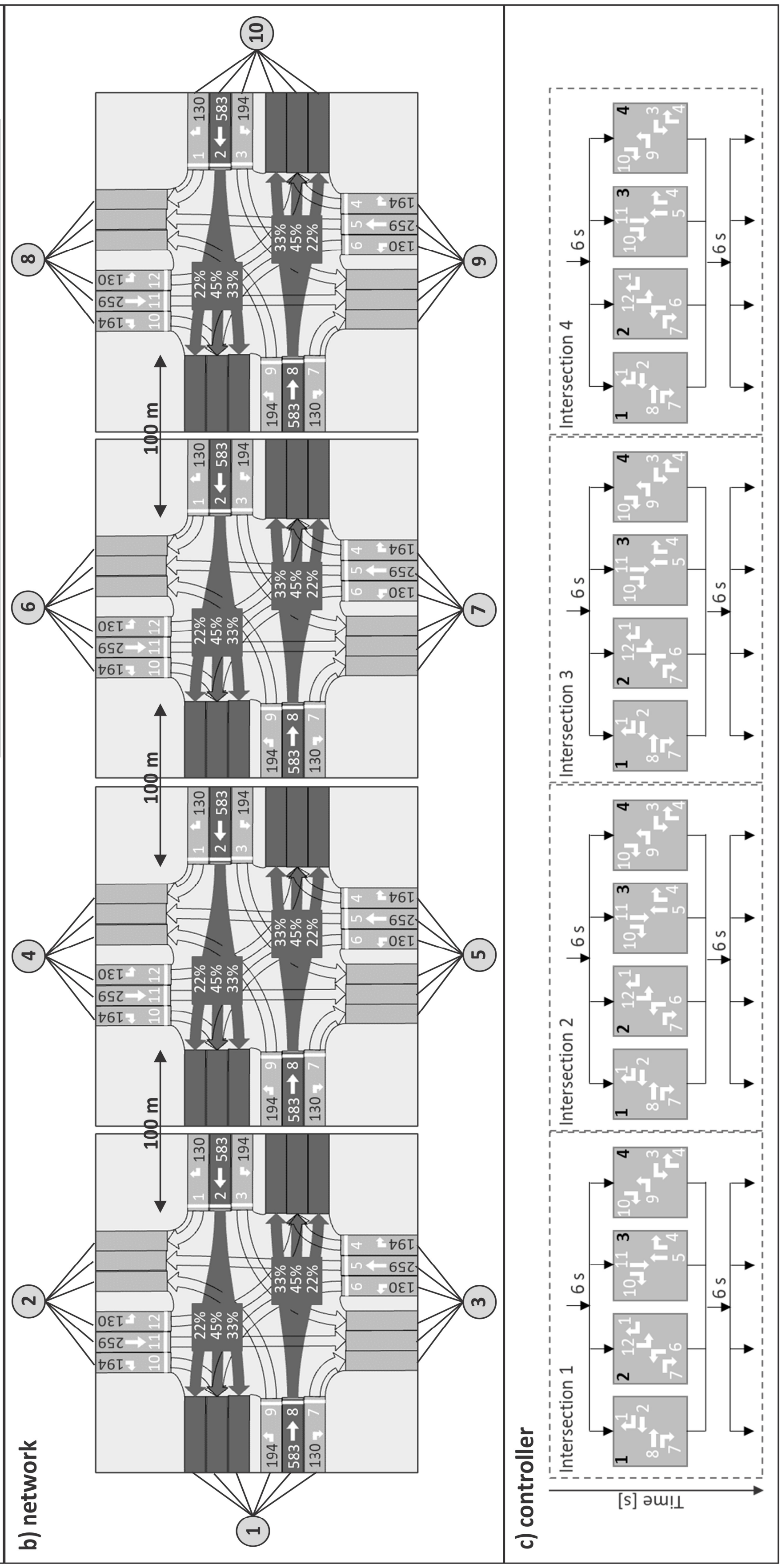


Figure 3.4: Traffic scenario with demand (a), network configuration (b), controller (c).

Demand pattern

The demand pattern of the traffic entering the corridor combines different crossing traffic flows to construct spillback effects (see Figure 3.4a and Figure 3.4b). The links on the main direction consist of traffic heading in the main direction as well as traffic heading for the side directions, which can be blocked by spillback on the main direction. The intersections are equally loaded. The underlying average traffic flows are given in the OD matrix [veh/h] (Figure 3.4a), and the resulting average loads [veh/h] and turn fractions are shown in the network (Figure 3.4b). A total demand period of 25 minutes is simulated (with an additional 20 minutes to assure the network empties). The total demand period is split into 5 periods of 5 minutes each, starting in the 1st period with a demand below average, increasing in the 2nd period to the average demand level, peaking in the 3rd period at a demand above average, decreasing again in the 4th period to the average demand level, and ending in the 5th period at a demand below average (Figure 3.4a). In this way, the overall demand follows a clock shape, starting at undersaturated conditions where queues easily dissolve, increasing to (over)saturated conditions where queues are building up and may cause spillback, decreasing to undersaturated conditions again where the traffic situation can recover.

In the simulation vehicle realizations are randomly generated. Each simulation is repeated 10 times with different random seeds to generalize results. These 10 random seeds remain the same for all the experiments to compare results for equal vehicle realizations. The random vehicle generation considers generation times as well as vehicle properties. The generation times between vehicles are exponentially distributed, with an average equivalent to the flow indicated in the OD matrix. The vehicle properties follow truncated normal distributions specified by the mean, standard deviation, and min-max values as given in Table 3.1a. Note that these vehicle properties are temporarily fixed (by setting the standard deviation to zero) in the reference situation to tune the parameters of the prediction model and to create the almost perfect world without prediction errors. The tuned parameter values of the macroscopic prediction model that correspond to the microscopic parameters of Table 3.1a are listed in Table 3.1b.

Table 3.1a: Microscopic vehicle properties.

Microscopic quantity*	Mean	Standard deviation	[Min, Max]
Simulation time step [s]	0.1	-	-
Reaction time [s]	0.8	0.1	[0.6, 1.0]
Reaction time at stop/traffic light [s]	1.6	0.2	[1.2, 2.0]
Maximum acceleration [m/s^2]	3.0	0.5	[2.0, 4.0]
Normal deceleration [m/s^2]	4.0	0.5	[3.0, 5.0]
Maximum deceleration [m/s^2]	6.0	0.5	[5.0, 7.0]
Maximum speed at link [m/s]	11.1	-	-
Speed acceptance factor	1.1	0.1	[0.9, 1.3]
Clearance in queue [m]	2.0	0.5	[1.0, 3.0]
Car length [m]	4.0	0.5	[3.0, 5.0]

* The microscopic parameters follow truncated normal distributions specified by the mean, standard deviation, and min-max values.

Table 3.1b: Tuned macroscopic quantities.

Macroscopic quantity		Tuned value
T	Time interval [s]	6.0
T_L	Loss time [s]	3.0
r_i	Saturation flow rate [veh/s] *	0.5
v_i	Speed at dedicated lane [m/s] **	8.3
L_i	Length of dedicated lane [m]	100.0
L_{VEH}	Distance headway in queue [m]	6.0
L_i/L_{VEH}	Storage capacity of lane [veh]	16.0
v_i^H	Propagation speed queue head [m/s]	4.0

* Note that the saturation rate is adjusted to 0,33 veh/s for the first two time intervals after green, including the effect of reaction times.

** Note that the tuned macroscopic free-flow speed is lower than the maximum speed in the micro-simulation, incorporating the travel time on the intersection surface. Note that a vehicle needs 12 s, i.e., $n_i = (L_i/v_i)/T = 2$ intervals, to travel from one intersection to the next in free-flow conditions.

3.3 Results

3.3.1 Reference: perfect world

Before looking into the behavior of the system performance in the (almost) perfect world (step I in Figure 3.2), the model match between the prediction model and the simulation model is addressed. Figure 3.5a shows the simulated and predicted delay over the simulation horizon for the 10 different demand realizations. The simulation horizon of 30 minutes is split into intervals of 1 minute, for each of which a prediction is made ($K=K^{UP}=10$). Note that a 60-seconds prediction horizon is representative for the values used in this study and therefore long enough to check the model match. The total predicted delay for the prediction horizon (in fact the objective of the control system (3.9)) is compared to the total simulated delay over this 1-minute period. In this way, for each demand realization 30 predictions are made, and the model match is checked over the entire simulation period, looking into undersaturated as well as (over)saturated conditions. Figure 3.5a clearly shows that the simulated delays follow the predicted delays. For each demand realization, the line of the simulated delay overlaps with the line of the predicted delay. To quantify the model match, in Figure 3.5b the simulated delays are plotted against the predicted delays and the R^2-value is calculated. The points almost perfectly lie on the diagonal with an R^2-value of 0.9996, indicating an almost perfect model match.

In this almost perfect world, where accurate predictions are made, the control system can reach its optimal performance. In Figure 3.6a, the system performance is presented as a function of the prediction horizon varying from 6 to 120 seconds (with a fixed update horizon of 12 seconds), and in Figure 3.6b, as a function of the update horizon, varying from 6 to 60 seconds (with a fixed prediction horizon of 60 seconds). The system performance, or in fact the system cost, is expressed as the average delay per vehicle on the corridor, a more intuitive representation of the total delay objective of the controller. The average delay is obtained by dividing the total delay of the entire simulation by the total number of vehicles in the simulation, i.e., a division by a constant number, since the simulation starts and ends in an empty network, affecting only the representation scale. Note that the average delay per vehicle is presented using a log-scale to focus on the lower delay values. The results of the 10 different simulation realizations are averaged, and the 95% confidence intervals are calculated to indicate the variation in the simulation results (together with min-max bandwidths that are presented on the background). Pairwise *t*-tests are performed to indicate whether the system costs are significantly changing for an increasing prediction horizon or update horizon, where a probability below 5% indicates a significant increase or decrease.

Figure 3.6a shows that increasing the prediction horizon improves the system performance in the corridor when perfect predictions are available. A short prediction horizon (6 seconds) considers only local traffic information nearby each individual intersection. This causes spillback and results in a high average delay of 313 [s] per vehicle. Increasing the prediction horizon allows looking ahead over multiple intersections, reduces spillback effects and decreases system costs up to an average delay of 85 [s] per vehicle. Note that the delay is not a smooth and strictly decreasing function. Figure 3.6a shows that there is a considerable bandwidth around the average results. The delay curve for each of the simulated realizations is fluctuating over the prediction horizons, particularly at the beginning of the curve for short horizons. Due to the finite nature of the prediction horizon and the inability of too short horizons to capture the whole process dynamics, each prediction horizon length results in a different sequence of optimization problems, a different sequence of optimal control plans and traffic states, and therefore in a different system performance. Moreover, the delay also varies considerably over the different demand realizations, causing spillback at different moments and with different severity. This is inherent to the (over)saturated traffic scenario with a high probability for spillback and due to the stochastic generation of individual vehicles. Averaging the realizations leads to less fluctuation in the system costs and a smoother curve. Pairwise *t*-tests indicate that the delay is a (piecewise) decreasing function (see Figure 3.6a). The delay significantly decreases with a large amount up to a horizon of 24 s, i.e., looking ahead 2 intersections considering a travel time of 12 s between intersections (Table 3.1b). The delay remains almost equal for a horizon of 30 s, and significantly decreases again, however with a smaller amount, for a horizon of 36 s, looking ahead 3 intersections. The system performance does not significantly improve anymore beyond this horizon for this corridor of 4 intersections.

Figure 3.6b shows that updating the decision more frequently using a shorter update horizon improves system performance only slightly when almost perfect predictions are available. A shorter update horizon results in more overlap in the subsequent decision sequences, shifting the prediction horizon more frequently with smaller steps, and gives more steady predictions and a better performing control system. When there is no overlap at all and the update horizon equals the prediction horizon of 60 s, consequences of decisions at the end of the update horizon are no longer considered, and the system costs significantly increase, however only gradually in absolute sense (increase in average delay from 85 [s] to 97 [s] per vehicle). The predictions are (almost) perfect and do reproduce the traffic situation well enough for longer prediction horizons, keeping the system performance steady, also for longer update horizons.

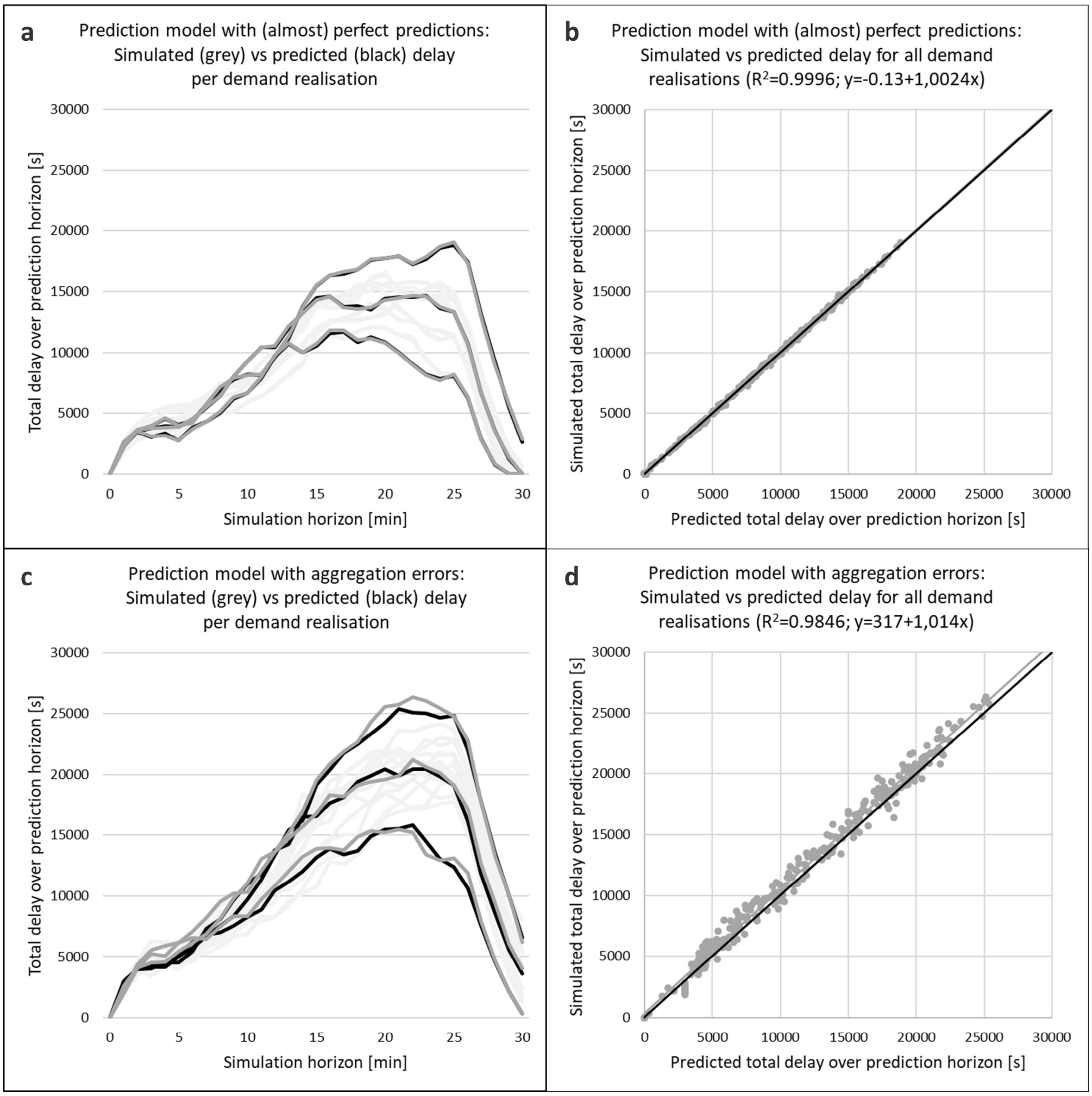


Figure 3.5: Model match between the microscopic simulation model and macroscopic prediction model for the reference 'almost perfect world' (top) and the presence of aggregation errors (bottom). The model match for all 10 simulation realizations is presented, in total as a quantitative R^2 analysis (right), and per realization as a function over the simulation horizon (left) where 3 realizations are highlighted.

3.3.2 Aggregation errors

Aggregating individual behavior in the prediction model (step II in Figure 3.2) introduces a model mismatch. Figure 3.5c shows that the line of the simulated delay no longer overlaps with the predicted delay, but that simulated delays deviate from predicted delays. However, the pair of lines that belong to the same run of prediction and simulation still lie close to each other and can be visually matched. The R^2 analysis in Figure 3.5d shows that the points no longer lie on the diagonal but are more scattered and lie slightly skew to the diagonal (the R^2 value decreases to 0.9846). The prediction model slightly underestimates the delay by assuming aggregated and uniformly distributed behavior and makes control decisions accordingly. In the simulation (as in real life) platoons of vehicles are formed, consisting of clustered vehicles heading for the same direction or arriving at the same time. These platoons do not match the control decisions based on aggregated uniform behavior and therefore lead to a higher and more varied delay than predicted.

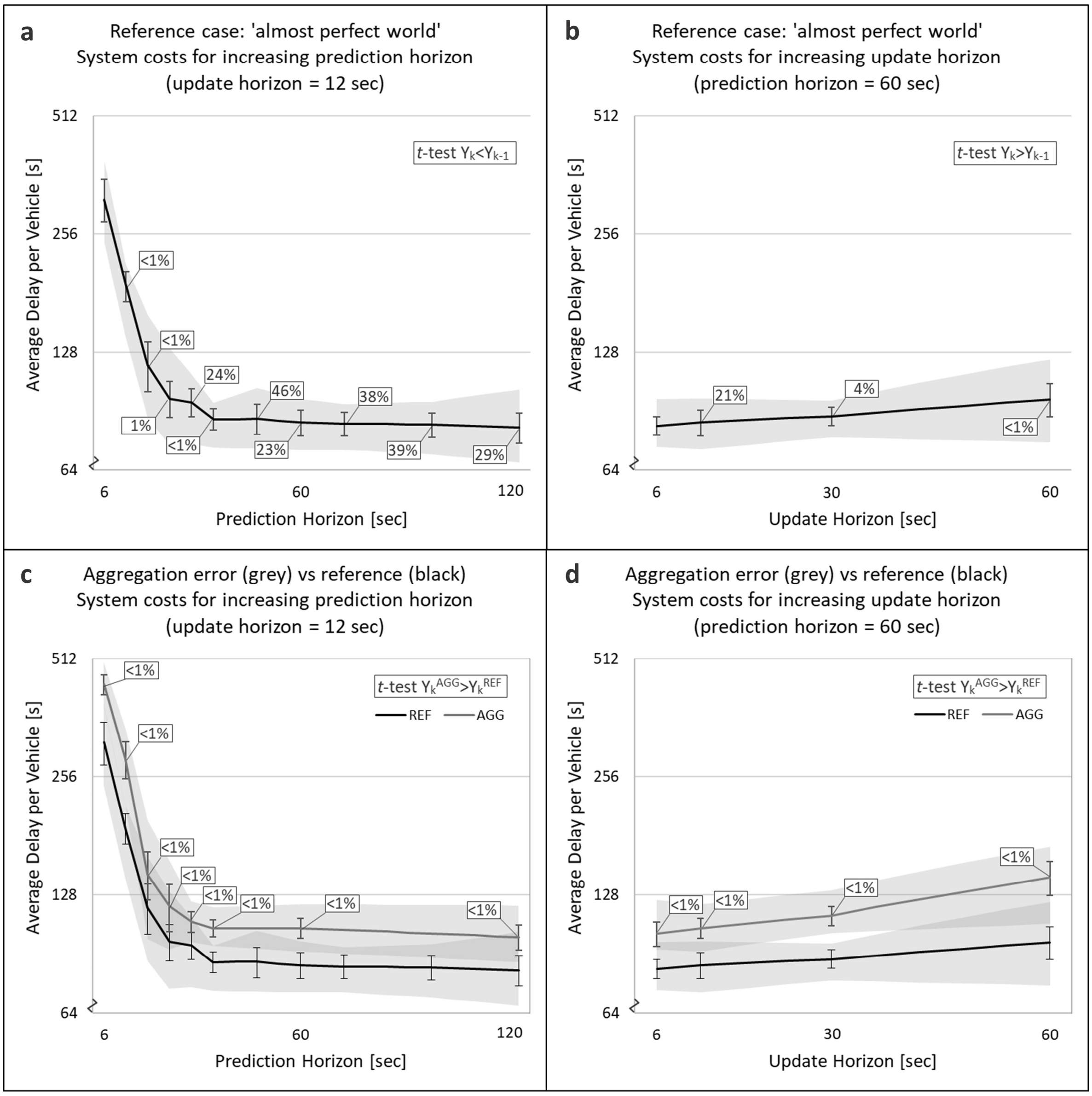


Figure 3.6: Performance of the controller with almost perfect predictions (top) and aggregation errors (bottom) for an increasing prediction horizon (left) and update horizon (right). Average values over 10 simulation realizations are presented, together with 95% confidence intervals (and min-max bandwidths on the background) to indicate variation, and *t*-values to indicate significant changes.

Despite the aggregation error in the prediction model, the predictive properties of the control system remain preserved. Figure 3.6c shows that the system costs still decrease, i.e., system performance still increases, for increasing prediction horizon. However, there is a significant performance loss compared to the reference situation with perfect predictions (see t-values in Figure 3.6c). The system cost increase from an average delay of 85 [s] to 105 [s] per vehicle. From a more detailed analysis it appears that 85% of the performance loss is caused by aggregating the individual destinations of vehicles into turn fractions, 10% is caused by aggregating individual arrivals into average demand, and 5% is caused by aggregating individual driving behavior. The aggregation into turn fractions is dominant for the simulated corridor, where many platoons are formed consisting of vehicles heading for the same direction. Platoons caused by clustered arriving vehicles (demand), or slower vehicles (driving behavior) play a lesser role in the simulated corridor, since the intersections lie close to each other. Moreover, these platoons affect the arriving times of vehicles in undersaturated conditions, and therefore have less influence in the simulated corridor with its many queues. Aggregated turn fractions cause the largest performance loss in saturated conditions. Accurate predictions of the individual turning directions of vehicles are most important in situations where the system is close to spillback from downstream movements. In these critical moments the aggregated prediction model with turn fractions is not accurate enough and causes significant performance loss.

The performance loss due to aggregation errors is clearly visible at large update horizons (see Figure 3.6d). For an update (and prediction) horizon of 60 seconds the system costs increase from an average delay of 97 [s] to 142 [s]. Note that for these settings the model mismatch is displayed in Figure 3.5. The performance loss can partly be compensated by decreasing the update horizon and reinitializing to the current state more often. However, a significant performance loss remains (see t-values in Figure 3.6d), leading to an increase in the average delay of 85 [s] to 105 [s] per vehicle. So, aggregating individual vehicle behavior in the prediction model results in a significant performance loss, or reasoning the other way around, including individual vehicle information in the predictive controller results in a considerable performance gain (provided that accurate predictions are available for individual vehicles, their destinations in particular).

3.3.3 Biases

Table 3.2 presents the system performance of the controller when there are additional biases in the aggregated prediction model (step III in Figure 3.2). For the different model quantities, the effect on the system costs of a range of relative biases is shown, i.e., e = -0.5, -0.2, -0.1, 0.1, 0.2, 0.5. Note that for each bias (table row) the average delay per vehicle can be graphically displayed as a function of the prediction horizon and as a function of the update horizon, in a similar way as for the reference and aggregation error in Figure 3.6. To get a total overview of the effect of all the biases, the essential points of the delay curves are listed in the table instead and the cell of each delay is colored according to its value. For comparison purposes also the system costs for the reference and aggregation error are listed (as displayed in Figure 3.6).

Table 3.2: System costs expressed in average delay per vehicle [s] for the predictive controller with biases, as a function of the prediction horizon (with fixed update horizon of 12 s) (left) and as a function of the update horizon (with fixed prediction horizon of 60 s) (right).

Legend												
Average delay per vehicle [s]	≤90	≤100	≤110	≤120	≤130	≤140	≤150	≤200	≤250	≤300	≤400	≤500	≤600

			prediction horizon [s]			update horizon [s]		
			12	**30**	**60**	**12**	**30**	**60**
Reference			189	95	85	85	88	97
Aggregation			284	110	105	105	113	142
Bias	Turn fractions (main direction) $\pi_{i\bar{i}} \in [0,1]$	-100%	271	109	102	102	123	163
		-50%	236	105	100	100	108	142
		-20%	267	107	104	104	107	132
		20%	325	127	114	114	128	161
		50%	368	158	129	129	145	200
		100%	396	286	234	234	301	353
	Queue head propagation speed v_i^H [m/s]	-75%	303	113	117	117	151	291
		-50%	314	118	118	118	140	159
		-20%	307	115	108	108	120	137
		20%	310	112	103	103	112	133
		50%	324	110	105	105	108	120
		75%	319	104	108	108	109	126
	Saturation flow rate r_i [veh/s]	-50%	607	461	275	275	314	416
		-20%	396	167	142	142	148	166
		-10%	323	127	116	116	127	157
		10%	224	103	100	100	107	127
		20%	190	103	102	102	108	131
		50%	210	118	115	115	124	156
	Travel time L_i/v_i [s]	-50%	292	167	149	149	146	155
		-20%	284	110	105	105	113	142
		-10%	284	110	105	105	113	142
		10%	284	110	105	105	113	142
		20%	284	110	105	105	113	142
		50%	326	105	100	100	106	141
	Demand Arrivals $a_i(k)$ [veh]	-50%	190	97	91	91	103	122
		-20%	202	103	99	99	107	120
		-10%	234	110	104	104	109	121
		10%	336	118	112	112	122	148
		20%	370	127	118	118	129	161
		50%	491	157	146	146	155	281
	State Queue $x_i(k_0)$ [veh]	-50%	248	153	139	139	132	136
		-20%	192	114	107	107	111	123
		-10%	263	110	100	100	110	133
		10%	332	118	106	106	121	150
		20%	339	118	111	111	122	149
		50%	414	174	141	141	142	198

Role of the prediction horizon

Although there is an added bias in the aggregated prediction model, the predictive properties of the control system remain preserved. The system costs still decrease for increasing prediction horizon (colors change from dark to light for increasing prediction horizon in Table 3.2). However, for some quantities, like turn fractions and saturation rate, for the larger biases of 50%, the prediction horizon needs to be increased to 60 s instead of 30 s to fully benefit from the lookahead capability. Note that in the experiments only one quantity is biased at a time and the other quantities contain accurate information. Apparently, despite the biases, there is enough structure in the predictions left to be beneficial for the control system. Overall, looking ahead and exchanging information from multiple intersections, even if the information contains biases that propagate through the network, leads to a better performance than considering only local information. Connecting intersections is of major importance at near saturated conditions to reduce the chance of spillback and improve the dissolving of queues.

Role of the update horizon

The update frequency plays an important role in reducing the performance loss caused by biases. Shortening the update horizon decreases the system costs (colors change from dark to light for shorter update horizons in Table 3.2). More frequently reinitializing to the actual state reduces the performance loss caused by biases. The reducing effect differs per quantity and error level. However, the performance loss is not compensated completely, i.e., if there is a performance loss due to the bias at an update horizon of 60 seconds, then there remains a performance loss at an update horizon of 12 seconds. Moreover, if the state itself is biased too much, the reinitialization to the state has no compensating effect anymore and/or the biased state itself causes an additional performance loss (see the -50% state bias in Table 3.2 where the performance loss due to aggregation errors is no longer reduced by shorter update horizons and the 50% state bias where an additional performance loss is measured due to the state bias on top of the performance loss due to aggregation errors).

Effect of a bias on system performance

In general, the biases in the prediction model result in a loss of system performance. The average delay per vehicle for the control system with biased predictions is mostly higher than for the control system with aggregation errors only (see Table 3.2, column for prediction horizon 60 seconds and update horizon 12 seconds). Sometimes there is no significant performance loss for the smaller error levels of ±10% and ±20%. Sometimes part of the performance loss caused by the aggregation error can even be compensated by introducing an additional bias, resulting in lower average delays. For example, an overestimation of the travel time partly compensates the aggregation error (see +50% travel time bias in Table 3.2). By aggregating the individual vehicle behavior, all vehicles are assumed to have the same speed in the prediction model. Consequently, the slowest vehicles miss the predicted green window and need to be given green twice, resulting in a performance loss caused by the aggregation error. By overestimating the travel time in the prediction, all vehicles become slower in the prediction model, and fewer vehicles will miss the predicted green window, regaining some of the performance. (Note that in this way there will be more vehicles that arrive slightly before the green window, but this delay is much smaller than when the window is missed.) Similar compensating effects of the performance loss due to aggregation errors can be noticed for other quantities. However, note that the average delays remain always higher than that for the reference situation, hence there is always a performance loss by introducing a bias compared to the almost perfect world without prediction errors.

For all quantities, a bias in one direction has a larger negative effect on the system performance than a bias in the opposite direction (colors are darker in one bias direction than in the opposite direction in Table 3.2). For example, an underestimation of the saturation rate causes a large performance loss, since the controller predicts too long and superfluous green periods to serve the traffic, reducing the capacity of the system. An overestimation of the saturation rate has less impact on the system performance. Since too short green periods are predicted, no predicted green is really wasted but given to other movements (and there is only a small capacity reduction by switching between movements more often). A small overestimation of the saturation rate may even be beneficial for the control system in (over)saturated conditions, since shorter green periods will be given to crossing movements, and the main direction on the corridor will get more gaps to dissolve its queues and reduce its spillback, compared to the control system with aggregated demands. Note that in undersaturated conditions this benefit disappears and there is a performance loss due to too short green periods and switching too often between movements. Similar effects can be noticed for the other quantities: the bias direction that clearly reduces the capacity of the control system causes the largest performance loss, and the other bias direction has less impact and may even be beneficial (compensate aggregation errors) for the control system in saturated conditions.

Ranking of quantities

Comparing all quantities, the saturation rate has the largest effect on the system performance in the simulated saturated corridor. For the smaller biases (up to -20%) the saturation rate already causes a significant performance loss, that cannot be compensated that well by reinitialization to the state more frequently. For larger biases (-50%) in the saturation rate the performance loss is clearly higher than for the other quantities. The high sensitivity for the saturation rate was also found in Chapter 2 for a single intersection. In a saturated corridor, inaccurate turn fractions also have a large effect on the system performance (note it was already the dominant factor for the performance loss due to aggregation). For a small additional bias (up to 20%), there already is a significant performance loss, however, the performance loss can be reduced quite well by reinitialization to the actual state. Accurate turn fractions are important at critical situations where the system is close to spillback from downstream movements. Reinitializing to the actual queue state more often, helps in predicting the available space downstream in these critical moments. For large biases (+100%) in the turn fractions, i.e., almost all traffic (twice the original 45% resulting in 90% of the traffic) heads for the main direction in the prediction (not an unrealistic assumption if direction information is missing), the prediction errors are too large to compensate and cause a huge performance loss. The biased demand, a structural error in the number of arriving vehicles, has less impact in the saturated corridor, since there are many queues in the network. The effect only becomes noticeable for larger biases (50%), when the predicted amount of crossing traffic becomes too large to find enough possibilities for the main direction of the corridor to dissolve its queues and reduce its spillback. Moreover, the demand bias can be compensated quite well by reinitialization to the actual queue state. The compensation between demand and queue information was also noticed in Chapter 2 for a single intersection. The travel time (or speed) has a relatively small effect in the saturated corridor. It becomes a more relevant quantity in undersaturated conditions in a network with larger distances between the intersections. The queue propagation speed has the smallest effect. Biases in the queue propagation speed cause temporary errors in the predicted storage space, that can be corrected well by reinitializing to the actual queue states. Only larger biases lead to a significant capacity reduction and performance loss. Considering this ranking, note that the state is still the most important quantity of them all, since it is needed to reduce the effect of the biases of the other quantities.

3.4 Discussion

The results of the sensitivity analysis show that predicting, by looking ahead over a horizon that corresponds to the propagation of traffic over multiple intersections, improves the system performance of the analyzed structure-free model predictive controller. This happens not only when perfect predictions are available but also when there are aggregation errors and biases in the system. This demonstrates the advantage of a global predictive network controller compared to local control in saturated networks. However, the sensitivity analysis also shows that there are conditions and guidelines that should be considered to take full advantage of the systems benefits. These will be discussed below.

The sensitivity analysis results show that increasing the prediction horizon increases system performance, in perfect conditions without prediction errors as well as in erroneous conditions with aggregation errors and biased predictions. There is an optimal horizon beyond which system performance does not improve anymore. This optimal horizon may be larger for a system containing biases, to fully benefit from the lookahead capability. In predictive control applications, a prediction horizon should be chosen that is large enough to look ahead in downstream (and upstream) direction over at least two intersections of a corridor, since connecting information on intersections is essential to improve system performance in saturated networks with spillback. The prediction horizon does not necessarily have to include (the traffic propagation over) all intersections on a corridor, since the performance gain becomes relatively smaller with the addition of each extra intersection. The exact choice of the prediction horizon can be expected to depend on the network configuration and should be larger if the distance between intersections is larger. However, a similar relation between system performance and increasing prediction horizon is expected as for the studied corridor.

The sensitivity analysis also shows that increasing the update frequency, which is the same as reinitializing to the current state more often, increases system performance and significantly reduces the effect of aggregation errors and biases in the control system. In structure-free predictive control applications, the update horizon should be set as small as possible (in units of seconds) to benefit from the highly adaptive property of the structure-free controller to correct mistakes quickly. Moreover, the accuracy of the initial state is essential to reduce the effect of prediction errors. If the initial state is biased too much, the performance gain of reinitializing to the state may be lost, and the effect of prediction errors may no longer be counteracted, as shown in the sensitivity analysis where aggregation errors are no longer compensated by reinitialization to a biased state. Therefore, the initial state is most important to predict accurately in the structure-free predictive control system.

In real life applications there is always a trade-off between the choice of the update horizon and the choice of the prediction horizon. If the update horizon is shortened to improve system performance by reinitializing to the current state more often, less calculation time is available to find an optimal control plan for the upcoming prediction horizon, for real-time operation. The prediction horizon should be decreased to limit the search space and to find an optimal solution in time, which may reduce the performance gain of the shortened update horizon. Or stated the other way around, if the prediction horizon is increased to improve system performance, more calculation time is needed to find an optimal control plan. So, the update horizon should be set larger, which may reduce the performance gain of the increased prediction horizon. This trade-off should be considered in the control application, depending on the available computational resources, particularly in large networks. In the studied corridor, however, this was not a dominant factor yet.

From the sensitivity analysis it turns out that aggregation errors in the prediction model cause a significant performance loss. The performance loss is expected to be even higher in real life, since there is more randomness in real life than in the simulation environment considered in this study. Note that the performance loss may be less for the more traditional cycle-based controllers, compared to the structure-free controller considered in this study, since cycle-based controllers make more aggregated decisions based on proportions of traffic. If a structure-free controller is used, individual information should be used, to fully benefit from the adaptive potential of the structure-free controller and significantly increase system performance. However, this individual vehicle information should be predicted carefully, otherwise biases can decrease performance again below the level of an aggregated system. The performance of future control applications can be improved significantly by using a structure-free controller and including individual vehicle information, in particular individual destinations, if the individual information is predicted accurately.

The sensitivity analysis shows that in general biases cause a significant additional performance loss (on top of the performance loss due to aggregation errors). Smaller biases (up to 20%) cause a proportional performance loss (up to 25%), which can partly be compensated by reinitializing to the actual state. Larger biases (50%) may result in a non-proportional performance loss (up to 150%), which can hardly be compensated by reinitializing to the actual state more frequently. Such large biases are not common in practice, except for quantities that are difficult to predict and are based on rough assumptions. An example that could occur is a lack of information on individual vehicle destinations or turn fractions for which it is assumed that all traffic is heading for the main direction. These rough assumptions resulting in large biases should be avoided. Note that in the sensitivity analysis the quantities have been disturbed one by one. In real life, biases of different quantities appear simultaneously and may amplify each other, resulting in more performance loss than indicated in this study. In the sensitivity analysis, a small bias could already cause significant performance loss for quantities that cannot be corrected by the initial state (for smaller update horizons) or by information on other quantities in the system (for larger prediction horizons). These model quantities are most important to predict accurately.

From the sensitivity analysis results, the saturation rate and the turn fractions appear to be the most sensitive quantities in the predictive control system and need to be predicted most accurately. However, these quantities are also most difficult to predict in real life, by lack of information and due to platooning (for example a slow vehicle decreasing the saturation rate for a whole platoon of following vehicles, or a platoon of vehicles with the same destination disturbing the average turn fractions). In the design of the control system, the prediction methods for these quantities, i.e., saturation rate and turn fractions, need to be improved and need to be made time-dependent, particularly in combination with individual vehicle information. Note that the quantities addressed in the sensitivity analysis in this chapter are the major components in a prediction model. A different macroscopic traffic flow model will change the exact results of the experiments but is expected to identify comparable model components as the most sensitive quantities in the control system. Note that some of the quantities, like travel time and queue head propagation speed may have more influence in network configurations with longer distances between intersections and other demand patterns. However, the saturation rate and the turn fractions are expected to remain the most sensitive quantities in saturated conditions since these quantities are essential in critical situations where the system is close to spillback.

For all quantities investigated in the sensitivity analysis, a bias in one direction has a larger negative effect on the system performance than a bias in the opposite direction. The bias direction that reduces the capacity of the intersections has the largest effect on the system performance in (over)saturated conditions. This bias direction should be avoided for all quantities, particularly if biases in different quantities amplify each other. In the design towards a more robust controller, one should slightly over/underestimate each quantity to the direction with the least impact on system performance. Moreover, adding a small deliberate bias in the least impact direction may even be beneficial for the control system, partly regaining some of the performance loss due to aggregation errors. As shown in the sensitivity analysis, adding a small bias to some of the quantities (such as the saturation flow) leads to a better performance than just taking the measured average. In (over)saturated conditions a small bias in a quantity in the right direction helps dissolving queues and reducing spillback, however, such a preventive bias may introduce an additional performance loss in undersaturated conditions. In the design of the control system, it should be considered how much performance loss is acceptable in many common situations compared to the huge amount of performance loss in fewer exceptional situations. This consideration deserves further research.

3.5 Conclusion

In this chapter a sensitivity analysis was set up and performed for a structure-free model-based predictive signal controller in a saturated urban corridor with spillback. In a simulation environment, the influence of prediction errors was studied on the system performance of the controller, as a function of the prediction horizon (lookahead capability) and update frequency (damping ability) of the control system. The results of the sensitivity analysis were translated into guidelines for the design of a structure-free model-based predictive controller, such that the highly adaptive and predictive system can better handle the effects of prediction errors in an (over)saturated network.

From the sensitivity analysis it can be concluded that the predictive property of the control system is strong and remains preserved under erroneous conditions. Increasing the prediction horizon from 0 to 60 seconds increases system performance up to 75% in the corridor when perfect predictions are available, but also when there are aggregation errors or even biases in the system, leaving enough remaining structure in the prediction model to rely on. This demonstrates the advantage of a global predictive network controller compared to local control in saturated networks. Looking ahead connects information of multiple intersections on arriving traffic (forward) and spillback (backward). The performance gain increases but flattens out with an increasing number of connected intersections. The gain is 40%, 70%, and 75% when looking ahead 1, 2, and 3 intersections in the simulated corridor. There is an optimal prediction horizon, beyond which the system performance does not improve anymore, however this horizon is not always easy to recognize due to the highly stochastic traffic process and may be longer for a system containing biases. This optimal choice of the prediction horizon is network dependent, but the behavior of the prediction horizon is expected to be similar as in the simulated corridor.

The update frequency plays an important role to reduce the effect of aggregation errors and biases. Decreasing the update horizon from 60 to 12 seconds in the corridor increases system performance up to 30%. Due to the highly adaptive property of the structure-free controller, mistakes can be quickly corrected by reinitializing to the actual traffic state more often. The sensitivity analysis has shown that this correcting ability can disappear if the initial state estimate contains biases, therefore the initial state is most important to estimate accurately in the structure-free predictive control system.

Although partly reduced, there remains a significant performance loss due to aggregation errors and biases. Aggregating individual vehicle behavior leads to a significant performance loss of 20% in the corridor, mainly caused by aggregating individual destinations into turn fractions, in a lesser extent also by aggregating individual arrivals and driving behavior. The latter may be larger in real life due to more stochastics and larger intersectional distances. A structure-free controller can adapt the control decision to closely match fluctuating arrival pattens and platooning. Therefore, there is a large benefit by including more detailed information on individual vehicle arrivals, in particular individual destinations, into the prediction model, provided that this type of information is available in real life and can be predicted accurately enough.

Additional performance losses due to biases were detected for all model quantities in the sensitivity analyses. Smaller biases (up to 20%) in the model quantities lead to proportional performance losses (up to 25%) in the corridor. Larger biases (of 50%) may result in an unproportioned performance loss (up to 150%) in the corridor and should therefore be avoided. The saturation rate is the most sensitive quantity in the control system in the saturated corridor, followed by the turn fractions already indicated as the dominant factor in the aggregation process. A bias in the saturation rate already causes a significant performance loss at the lower error levels that is difficult to compensate by reinitialization to the actual state. Therefore, the saturation rate is the most important model quantity to predict accurately.

The sensitivity analysis also shows that for all model quantities one direction of the bias has more impact on the system performance than the other direction. The bias direction that reduces the capacity of the intersections has a large effect on the system performance in (over)saturated conditions and should be avoided for all quantities. A bias in the other direction is less severe and (in small amount) may even be beneficial to the system, regaining some (up to 10%) of the performance loss due to aggregation errors. In (over)saturated conditions a small bias in a quantity in the right direction helps dissolving queues and reducing spillback, however, may introduce an additional performance loss in undersaturated conditions. This consideration is left for further research towards more robust control systems.

Overall, considering these insights and guidelines, a structure-free model-based predictive controller can be designed to function adequately under erroneous conditions. Moreover, the adaptive ability makes the structure-free predictive controller promising for future applications, where information on individual vehicles becomes available.

Chapter 4

Robust approach to reduce the impact of prediction uncertainties in network control

Robust control approaches are used in fixed-time traffic signal control to protect against performance loss due to large demand uncertainties. However, given the time-consuming optimization process, robust control is not applied much yet in real-time adaptive predictive traffic signal control, to protect against performance loss caused by all kinds of prediction uncertainties. Therefore, in this chapter, a robust scenario-based minmax optimization is applied to a structure-free model-based predictive traffic signal controller to reduce performance loss due to uncertainties in prediction model parameters, a dominant factor in these highly adaptive systems. Besides, a heuristic robust approach is proposed that determines a subset of relevant worst-case scenarios for the minmax optimization, saving calculation time for real-time applications. The robust controller is analyzed for a critical traffic situation, a saturated corridor of intersections with a high chance of spillback during peak periods with uncertain turn fractions. The analysis shows that the robust predictive controller reduces performance loss considerably, up to 60%, in the worst-case parameter scenarios. The heuristic approach reduces the scenario set to 12% of its full size in the robust optimization, saving calculation time while maintaining a similar control performance. Furthermore, applying the robust controller only during the uncertain peak periods, increases the average system performance over all scenarios for the entire period, being only temporarily overprotective for the better good. Overall, the heuristic robust controller provides protection against performance loss due to prediction uncertainties and reduces calculation times, making robust control more applicable for real-time adaptive predictive traffic signal control.

This chapter is based on the paper: M. C. Poelman, A. Hegyi, A. Verbraeck, and J. W. C. van Lint, "Robust approach for structure-free model-based predictive traffic signal control considering uncertainties in model parameters," *Under review*.

4.1 Introduction

Over the last years, highly adaptive predictive traffic signal control approaches are emerging due to new technology for communicating vehicles and real-time data availability on individual vehicle movements. These adaptive signal control systems make it possible to anticipate in real time to highly fluctuating demand patterns and individual vehicle arrivals, improving the control performance considerably. For an overview of the development of traffic signal control methods over the years see [1], [11], [5]. Model-based predictive control is an adaptive approach that pro-actively controls traffic in a network [19], [24]. The control decision is repeatedly optimized based on model-based predictions of the evolution of expected future traffic states in the network. A rolling horizon approach is applied. The decision sequence for the upcoming prediction horizon is optimized, after which the prediction horizon is shifted, and a new control decision sequence is determined starting in the reinitialized current traffic state. Most of the existing model-based predictive control methods [19], [24] use a pre-defined cyclic control structure, optimizing cycle times and green splits only, based on average predicted traffic flows in a cycle. Advanced predictive control systems have more degrees of freedom and a higher adaptivity level [11], optimizing the order and combination of the signals as well and reconsidering signal settings more frequently (each few seconds). These structure-free (non-cyclic) predictive controllers essentially can anticipate more quickly to changing traffic patterns and can match the fluctuations in the demand pattern better up to the individual vehicle level, improving the performance of the control system. Overall, structure-free (non-cyclic) model-based predictive signal controllers have a high potential controlling the traffic system in an efficient manner. However, the existing model predictive control methods mostly consider perfect predictions, whereas in real life predictions errors arise due to uncertain demand input or uncertain initial states, and due to model mismatch, i.e., uncertain model relations and parameters. The performance of structure-free model-based predictive control systems can be sensitive to these prediction errors, resulting in high performance loss in critical traffic conditions, as was shown in Chapter 2 and 3. Especially model mismatches have a more dominant influence on the control performance of these highly adaptive systems (see Chapter 3), since large demand fluctuations are already covered by design. Therefore, this chapter explicitly considers prediction errors in structure-free model-based predictive traffic signal control due to uncertain model parameters by applying robust control techniques in this highly adaptive real-time system.

In existing literature, robust control is initially and mostly applied to fixed-time control systems. Fixed-time controllers are designed offline based on an average traffic demand. Robust approaches are used to protect against high performance loss due to large day-to-day demand fluctuations that were not considered in the original control design (for an overview see [31], [32]). These robust offline fixed-time control methods are developed as an alternative for real-time adaptive control methods that need a more complex infrastructure to adapt control plans in real time based on the actual traffic demand. Margins are included in the offline control plan instead, anticipating on unforeseen demands and reducing performance loss in these worst-case scenarios, without becoming overprotective and increasing the system performance for the average demand scenario too much. Different robust control approaches can be distinguished in the way demand uncertainties are included, varying from considering complete demand distributions, using (random) demand scenarios, to considering demand (minmax) boundaries only (see Section 4.2 for more details). Note that strictly speaking the term robust programming is reserved for boundary-based methods that minimize performance loss in worst-case scenarios only, by a so-called minmax optimization. The terms stochastic and distributional programming are used for scenario- and distribution-based methods, that compare, weigh, and average performance loss over multiple demand scenarios and realizations in the optimization process. Often, also in this thesis, the term "robust control" is used more generally to indicate all approaches that explicitly consider uncertainties in the control optimization to reduce their effect on the control performance.

In some more recent work, robust control is applied to adaptive control systems as well, mostly in a cyclic model-based predictive control system (see [34], [29], [30]). Although adaptive control anticipates to large demand fluctuations by design, still demand errors in smaller time-varying fluctuations are left in the system that may influence the control performance. Robust control is applied to protect against performance loss due to these smaller time-varying, mostly cycle-to-cycle, demand fluctuations and corresponding uncertainties in the amount of traffic in the predicted states. Similar approaches as for fixed-time robust control can be distinguished for robust adaptive control (see Section 4.2 for more details), which all explicitly consider uncertainties in the control optimization to reduce their effect on the control performance.

However, existing work on robust real-time adaptive control is still rather limited, probably due to the problem complexity and large computation times of robust control techniques (see [31], [32]). Besides, so far, robust control approaches only consider cyclic control with continuous green split variables, and do not consider structure-free control yet with discrete binary red-green variables that increase problem complexity further. Moreover, existing robust control approaches mainly focus on demand uncertainties as the external disturbances, which have the largest influence on system performance in fixed-time control (see [31], [32]) and to a lesser extent in cyclic adaptive model predictive control (see [34], [29], [30]). Whereas other type of uncertainties, e.g., in the prediction model relations and parameters, may rise in adaptive systems that influence control performance more dominantly since demand fluctuations are better anticipated by the increasing adaptivity level of the controllers. In a highly adaptive structure-free model-based predictive control system uncertainties in the parameters of the prediction model have indeed significant influence on the system performance (as shown in Chapter 2 and 3). Therefore, robust control techniques become interesting for these highly adaptive structure-free model-based predictive control systems to protect against performance loss due to the uncertainty in model parameters.

Hence, this chapter studies robust control techniques in <u>real-time adaptive</u> and <u>structure-free</u> model-based predictive signal control to protect against performance loss due to <u>uncertainties in model parameters</u>. First the robust control techniques are outlined for the existing robust offline fixed-time control methods and the existing extensions to robust real-time adaptive cyclic control methods are described, which protect against performance loss due to demand fluctuations. Then the robust control approach as studied in this chapter is defined in its new context, i.e., in a real-time model-based predictive control system with a highly adaptive structure-free controller with uncertain parameters in the prediction model. Furthermore, practical drawbacks are considered when applying robust control in such a highly adaptive and real-time context. The rolling prediction horizon of the model-based predictive control system is limited, and the full consequences of a decision may fall outside the scope of the predictive controller. The robust controller must identify the worst-case scenarios in time during this limited time horizon, to protect against a possible performance loss later in time. Besides, due to the high update frequency of the control decision sequence in a highly adaptive structure-free control system, each few seconds instead of each cycle time as in traditional cyclic adaptive control, there is only a very limited amount of computation time available to evaluate all uncertainty scenarios. Therefore, a heuristic approach is suggested that needs to evaluate only a limited number of scenarios, constructing a subset of relevant scenarios on the fly during the rolling horizon process of the real-time controller that most likely contains the worst-case scenarios. In a simulation environment, the robust real-time adaptive structure-free model-based predictive signal controller is applied to a critical traffic situation, i.e., a corridor with high risk of spillback, where wrong assumptions on the uncertain model parameters may result in high performance losses. Turn fractions are chosen as the uncertain parameters, since these are one of the most sensitive parameters in such a corridor with spillback (see Chapter 3). The example case shows the potential of robust techniques for real-time structure-free model-based predictive signal control, in terms of control performance, by explicitly considering uncertainty in model parameters.

4.2 Background

4.2.1 Robust offline fixed-time control

Robust traffic signal control is initially applied to fixed-time controllers. Fixed-time controllers are designed offline with pre-defined signal control plans based on an average traffic demand. Robust approaches are used to protect against high performance loss due to large day-to-day demand fluctuations that were not considered in the original control design. For an overview of the first robust fixed-time control approaches see [31], [32]. These robust fixed-time traffic signal controllers are developed as an alternative for adaptive control methods that need a more complex infrastructure to detect the actual traffic pattern and adapt the traffic signals accordingly in real time. In the fixed-time traffic signal controllers, robust approaches are used instead to build in margins in the offline control plan such that, once the control plan is active, a deviation to the average demand pattern can still be controlled adequately preventing a worst-case scenario with unforeseen and unproportionally large performance loss. In the last two decades, different robust fixed-time controllers have been designed including enough margin in the control plan to protect against performance loss in these worst-case scenarios, without becoming overprotective, i.e., without decreasing the control performance too much for the average demand pattern.

The considered literature on robust approaches in traffic signal controllers is displayed and classified in Table 4.1. In all the considered robust approaches for fixed-time control (indicated in non-bold black in Table 4.1), as the base, an optimal controller with a pre-defined cyclic structure is used, i.e., the order and combination in which movements get green are fixed. The cycle time and the green times of the different movements, and offsets in case of coordinated control, are determined offline by optimizing the control performance, mostly expressed in terms of delay, assuming an average and static demand pattern. The robust approaches differ in the way demand uncertainties are included in the optimization process of the fixed-time controller, regarding:

(i) Available information on demand uncertainties.
The robust approaches make different assumptions on the available information on the uncertainties in the demand, varying from no or little information to full information. In the basic fixed-time controller, no additional uncertainty in the average demand is considered (see first remark in Table 4.1). The least and easiest (best available) information level needed in a robust approach is the upper and lower bounds of the demand, which already makes it possible to identify worst-case scenarios, as shown in [31], [32]. Then, as in most of the considered approaches, a representative set of specified or randomly drawn demand scenarios is assumed, to not only identify the worst-case scenario, but also to weigh the worst-case performance with respect to the performance of the other scenarios using the chance of occurrence, e.g., as applied in [31], [38], [39], [40], [41], [42], [43]. In the most detailed approaches, the complete distribution of the demand is assumed to be known, to be able to include, weigh, and compare all the effects on the control performance of all possible demand realizations in the control optimization, e.g., as applied in [44], [45]. Note that there are even approaches that consider sets of distributions, assuming the real demand distribution is unknown but should be part of a wider range (see [46]).

(ii) Relation between demand uncertainties and delay (system performance).
Different functions are used to translate the value or distribution of the demand into the value or distribution of the delay, i.e., system performance, for a given candidate control plan. In fixed-time control for a single intersection, often a static analytical delay function is used, like a Webster or HCM delay function (see [31], [32], [44], [43]). Traffic models are used as well for delay evaluation in robust fixed-time control to consider a more detailed evolution of the traffic over time, useful in case of peak demand shapes, and place, useful in case of coordinated network control. The applied traffic models are varying from macroscopic dynamic models, like a store-and-forward approach or more advanced models including queue dynamics and shockwaves, e.g., the model used in [45], LWR model in [46], and CTM model in [38], [39], to the more detailed microscopic simulation models that include individual vehicle dynamics and stochastics, e.g., as applied in [40], [41] and [42] that combines microscopic simulation with a metamodel. Note that the analytical relations, like the static delay functions and closed-form model formulations (as applied in [45], [46]), are more suitable for the full distribution approach, and the simulation models, macroscopic as well as microscopic, are more applied in a scenario-based approach (see Table 4.1). Note that under symmetric demand uncertainties, the delay distribution becomes asymmetric, since higher demands result in unproportionally higher delays.

(iii) Objective function of the delay optimization under demand uncertainties.
The robust approaches use different objectives to reduce the effect of demand uncertainties on the control performance expressed in terms of delay. Note that the basic fixed-time controller, optimizes the average delay related to the average demand. In a similar way, in the robust control approaches, the expected value of the mean delay can be optimized, considering the asymmetric delay relation (ii), to aim for a better average control performance under demand uncertainties, as applied in [44], [45], [40], [41]. The expected

value of the variance of the delay can be optimized as well, solely or in combination with the mean delay, to reduce the fluctuation in the control performance to both sides, reducing high as well as low delays, as applied in [31], [42]. Optimizing the expected value of the exceeding of the mean delay reduces only the high delays, focusing on the negative consequences of demand uncertainties, as applied as a percentile objective in [39] and an expected value at risk minimization in [31], [38]. In the strictest formulation of robust control, the delay of the worst-case scenario is optimized following a minmax optimization approach, as applied in [31], [32] and as used in [43], [46] in combination with expected mean optimization. Note that the minmax formulation guarantees an upper bound for the delay in worst-case scenarios, whereas expected value optimization only reduces the chances on high delays. Note that the minmax optimization only needs the range of demand uncertainties to identify the worst-case scenario with highest delay, whereas the expected value optimization needs more information on the distribution of the demand uncertainties, in discrete scenario-based or continuous form, to weigh the delay of the different demand realizations with the chance of occurrence. Hence, the choice of the objective function to reduce the effect of demand uncertainties on the control performance strongly depends on the available information on the demand uncertainties (i).

Table 4.1: Classification of the considered robust approaches for robust offline fixed-time control (black), robust semi real-time adaptive control (bold grey), and robust full real-time adaptive control (bold black).

		delay function			
		static delay function	macroscopic dynamic model		microscopic dyn. model
demand uncertainties	**delay objective**	Webster, HCM	no queuing store-forward	inc. queuing CTM, LWR	microscopic simulation
average demand no uncertainties	MIN average delay	[47]*, **[48]***			
demand boundaries	MIN MAX delay	[31], [32]	**[34]**, **[29]**		
demand scenarios	MIN MAX delay	[43]		**[49]****	
	MIN expected delay mean/variance/exceeding	[31], [43]	**[50]**, **[51]**, **[30]**	[38], [39], **[52]**	[40], [41], [42]***
full demand distribution	MIN expected delay mean/variance/exceeding	[44]		[45]	
set of demand distributions	MIN MAX expected delay mean/variance/exceeding			**[46]**	

* Although the traditional (Webster-based) fixed time controllers consider no additional uncertainties around the average demand, uncertainties in the demand pattern due to fluctuations in arrival times of individual vehicles (Poisson process) are considered, resulting in a basic robust margin.

** Freeway ramp metering control application, all other robust approaches are urban signal control applications.

*** Microscopic simulation with a tuned metamodel to reduce calculation times.

The different robust approaches use a different combination of choices made in (i), (ii), and (iii), resulting in different methods with the same common goal "optimizing the control performance under demand uncertainties". Note that strictly speaking the term "robust programming" is reserved for the worst-case optimization, i.e., min-max formulation in (iii) with demand borders in (ii). And the term "stochastic programming" is used for expected value optimization in (iii) for randomly drawn demand scenarios in (ii). As a combination, robust min-max programming can be applied in a scenario-based approach as well. The more general terms "robust control" or "robust control approaches" are used to indicate the overall set of control methods that reduce the effect of demand uncertainties on the control performance.

The existing robust approaches for fixed-time traffic signal control (Table 4.1) aim at finding the right balance between the average and the worst-case system performance, such that high delays are prevented in worst-case demand scenarios and the delay is not increasing too much in the average demand scenario (no overprotectiveness). Note that the problem complexity of fixed-time control increases considerably using robust approaches. However, computation times are not much of a concern in robust fixed-time control since the control optimization is offline. Instead, existing approaches aim at finding the exact solution in the control optimization to guarantee optimality, in robust minmax programming in particular, to guarantee an upper bound for the delay in worst-case scenarios.

4.2.2 Robust real-time adaptive control

Robust control approaches can also be found for real-time adaptive signal control, however, the existing literature is limited. Control optimization under uncertainties is a time-consuming process, which becomes more difficult to apply in real time. Therefore, in-between solutions are suggested using pre-defined offline robust control plans in a real-time application (indicated in boldface grey in Table 4.1). These approaches determine fixed-time robust control plans offline and switch online between these control plans by a decision rule [46], [48], or slightly adapt the control plans [52], [50]. These robust semi real-time adaptive control methods will not be discussed further. So far, only a few approaches fully determine the robust signal control program online in real time, mainly applied in a model-based predictive control system considering cyclic control (indicated in boldface black in Table 4.1), from which [49] is a freeway ramp metering application, and [34], [29], [30], [51] are urban signal control applications. Note that real-time adaptive control already anticipates to the large day-to-day demand fluctuations by design, however, smaller cycle-to-cycle demand uncertainties remain that may have less but still significant influence on the control performance. Therefore, robust real-time adaptive control aims at reducing the effect of these cycle-to-cycle demand uncertainties on the system performance, preventing a large performance loss in the worst-case demand scenarios, and maintaining a reasonable performance in the average demand scenarios.

In robust real-time adaptive control similar approaches are followed as in fixed-time robust control (Section 4.2.1), however now they are applied in a real-time model-based predictive control system. An optimal controller with a cyclic structure is still used as the base. The cycle time, green-times, and offsets of the control plan are determined in real time by predicting the evolution of the expected future traffic states in the network and optimizing the total system performance. A rolling horizon approach is applied, in which the control plan is repeatedly optimized for the upcoming prediction horizon, after which the prediction horizon is shifted (see [19], [24] for an overview of model predictive control).

In robust real-time adaptive control methods (see Table 4.1 indicated in boldface black) similar choices can be made regarding (i), (ii), (iii), however in real-time control these choices are restricted by the available calculation time defined by the update frequency of the model-based predictive control system. (i) Note that the average demand pattern is no longer static, but time dependent and fluctuating over cycles. The considered information on the demand uncertainties around this time-dependent demand pattern should be sufficient to identify worst-case scenarios and to roughly compare scenarios but should not be too time-consuming to evaluate. Therefore scenario-based methods [30], [51], [49] and boundary-based methods [34], [29] are popular approaches in real-time robust control. (ii) A dynamic traffic flow relation is needed to predict the traffic state evolution in the network over time and place, and to translate the demand uncertainties into predicted traffic state uncertainties and corresponding effects on the system performance, mostly expressed in delay. A microscopic model is too time-consuming. Therefore, macroscopic dynamic models are used with no or simplified queueing, like the Store-and-Forward model in [34], [29], [30], [51]. (iii) All the different objectives to reduce the effect of demand uncertainties in the control performance can be applied in robust real-time adaptive control, dependent on the available information on the demand uncertainties, like the expected mean optimization as applied for the scenario-based methods [30], [51], and the minmax optimization as applied for the boundary-based method [34], [29] and scenario-based method [49].

Robust real-time adaptive control approaches, like fixed-time robust control approaches, aim at finding a right balance between the average and the worst-case system performance. Since problem complexity is an issue in robust real-time control, existing literature also addresses parameterization to limit the number of control variables [49], and heuristic solution methods [30] in combination with learning methods to speed up the delay evaluation [30]. Optimality cannot always be guaranteed in robust real-time adaptive control approaches since an exact solution cannot always be obtained in time. Therefore, in the case of robust min-max programming, the upper bound for the delay in worst-case scenarios cannot always be guaranteed either.

Existing work on robust real-time adaptive signal control is still rather limited, and mainly applied in a model predictive control context (as discussed above). Besides, so far robust control is only applied to cyclic control, in which the order and combination in which the movements get green is fixed and continuous variables, like cycle times, green times, and off-sets, are optimized. Robustness in structure-free control, where movements can have green in an arbitrary order and combination (no imposed cycles), has not been considered yet, probably since structure-free control with {0,1} control variables is already a complex integer programming problem, which becomes even more complex after introducing uncertainties, which makes it even harder to solve, especially in real time. Note that [45], [46] mention the possible use of {0,1} control variables in offline robust approaches, however still present a cyclic constraint example. Moreover, so far robust control focusses mainly on uncertainties in the demand, which has the largest influence on system performance in fixed-time control and to a lesser extent in cyclic real-time adaptive control. Robust approaches for other types of uncertainties are limited, only a few examples in model-based predictive control can be found, [34], [29], [30] consider additional disturbances in the amount of traffic in the systems predicted states, e.g., due to parking and uncontrolled junctions, [51] considers uncertain departures, i.e., saturation rates, restricted to local intersection control, and [49] mentions the possibility to consider model uncertainties, like turn rates, but does not include them in the example.

Highly adaptive structure-free predictive signal controllers are emerging due to new technologies [11], [5], which makes it possible to pro-actively adapt traffic signals in an arbitrary order and combination to highly fluctuating demand patterns up to individual vehicle arrivals. The more adaptive the controller becomes, the more fluctuations in the demand are anticipated by design, and other uncertainties in the predictive control process become more dominant in the system performance. Uncertainties in the parameters of a prediction model can have significant influence on the control performance of such a structure-free model-based predictive controller (as was shown in Chapter 2 and 3). Therefore, robust approaches for these structure-free model-based predictive controllers become interesting to protect against performance loss due to all kinds of model uncertainties and prediction errors. Hence, this chapter studies the robust control principles (as discussed in this section) in real-time adaptive and structure-free model-based predictive signal control to protect against performance loss due to uncertainties in model parameters.

4.3 Approach

4.3.1 Robust real-time structure-free model-based predictive control

In this chapter, a robust real-time structure-free model-based predictive traffic signal controller with uncertain model parameters is defined and analyzed in a simulation environment. The ideas of robust control, as explained in Section 4.2, are applied in a real-time model predictive control context, in which the control decision for the upcoming time horizon is repeatedly optimized based on a model-based prediction of the expected future traffic states in the network. The controller is assumed to be structure-free, i.e., signals can be activated in an arbitrary order and combination and there are no imposed cycles, and, therefore, the controller can pro-actively adapt to highly fluctuating demand patterns up to individual vehicle arrivals. Since the upcoming demand fluctuations are anticipated by the highly adaptive structure-free model-based predictive controller, uncertainties in the assumptions of the prediction model itself become more dominant in the control performance. Therefore, robust control approaches are applied to protect against possible performance loss due to these parameter uncertainties in the prediction model.

The robust controller should protect against performance loss in critical traffic situations, where uncertainties (in this case in the parameter values of the prediction model) may have large consequences in terms of control performance. The effect of a parameter uncertainty may not be visible immediately but can have unproportionally large performance consequences later in time. The real-time predictive controller has to evaluate all the possible parameter scenarios on the fly and should recognize these worst-case parameter scenarios in time to build in a safety margin to prevent such a large performance loss. However, this is a difficult task in a real-time adaptive context, where there is only a limited amount of computation time available to calculate a new decision dependent on the update frequency of the controller. The prediction horizon of a real-time model-based predictive controller is limited, due to limited amount of calculation time, and the full consequences of parameter uncertainties may fall outside the scope of the controller. Furthermore, the available amount of calculation time may be too small to evaluate all uncertainty scenarios, especially in a highly adaptive structure-free control system with a new decision optimization each few seconds. Besides, even if the worst-case parameter scenarios can be identified in time, the control system should not become overprotective, and should not only reduce performance loss in worst-case scenarios but should maintain a reasonable average system performance as well (as explained in Section 4.2). In this chapter, the following aspects of the robust controller are addressed and analyzed in more detail:

1. Limitation of the optimal (non-robust) controller in real time.
   As a reference, a real-time structure-free model-based predictive controller is defined based on average parameter settings. In a simulation environment, it is shown that this real-time optimal (non-robust) controller, without considering uncertainties in the parameters at all, may result in high performance loss in worst-case scenarios.

2A. Potential of full robust control protecting against the worst-case scenarios in real time.
   The robust version of the real-time controller is defined using a worst-case (minmax) optimization over all parameter uncertainty scenarios. The prediction horizon is assumed to be large enough to include the main effects of the parameter uncertainties. The complete set of parameter uncertainty scenarios is assumed to be known and assumed to be fully evaluated in the optimization, as if unlimited calculation time is available. The extent to which the robust controller can reduce performance loss in worst-case scenarios is analyzed. This shows the potential of the full robust controller in real time.

2B. Identification of the relevant worst-case scenarios in real time (heuristic robust approach).
   A heuristic robust approach is suggested that needs to evaluate only a limited number of scenarios in the optimization, constructing a subset of relevant scenarios on the fly during the rolling horizon process of the real-time controller that most likely contains the worst-case scenarios (Section 4.3.2). The heuristic approach saves calculation times and makes robust control more applicable in real time. The extent to which the heuristic approach can still reduce performance loss in worst-case scenarios is analyzed.

3. Overprotectiveness.
   The real-time robust controller should take enough margin on the fly to prevent high performance loss in worst-case scenarios, but not too much, otherwise the controller becomes overprotective decreasing the average system performance. To study overprotectiveness, the robust worst-case (minmax) optimization is compared to an optimization of the average (mean) system performance over all parameter scenarios.

Hence, a robust real-time structure-free model-based predictive signal controller is defined and tested in a simulated critical traffic situation, to analyze if worst-case parameter scenarios can be recognized in time, from a relevant subset of parameter scenarios, and performance loss can be reduced without becoming too overprotective.

In Section 4.3.2, the mathematical formulation of robust parameter settings in the structure-free model-based predictive control system is presented. First the optimal controller is described mathematically, then the robust version of the mathematical problem is introduced, and the heuristic robust approach is outlined. Note that, following the classification in Section 4.2, (i) a scenario-based approach is followed to define the parameter uncertainties, (ii) a macroscopic dynamic prediction model, i.e., a store-and-forward with additional queueing, is used to determine the effect of parameter uncertainties on the control performance expressed in delay, and (iii) a minmax optimization is applied to reduce the effect of parameter uncertainties on the control performance. In Section 4.3.3, the simulated traffic case is outlined, used to analyze the extent to which the robust principles protect in a real-time context against high performance loss due to uncertain model parameters.

4.3.2 Mathematical formulation robust control system

Optimal control

To define the optimal structure-free model-based predictive controller, the continuous time horizon is split into discrete control intervals, i.e., $[(k\text{-}1)T, kT)$, with k the discrete time index referring to the k[th] interval and T [s] the duration of the interval. At each control interval the signal settings at the intersections in the network may change. All movements i at the intersections can be controlled separately, however (without loss of generality) will be controlled together in pre-defined movement groups j_m , i.e., set $I(j_m)$ of non-conflicting movements $i \in I(j_m)$ at the intersection m. Signal states $s_i(k) \in \{0(\text{red}), 1(\text{green})\}$ and movement group states $p_{j_m}(k) \in \{0(\text{red}), 1(\text{green})\}$ are introduced for all movements i, movement groups j_m at intersections m, and time indices k. Note that only one movement group per intersection can be active at a time, i.e., $\sum_{j_m} p_{j_m}(k) = 1 \; \forall m \; \forall k$, and the signal states follow the movement group state, i.e., $s_i(k) = p_{j_m}(k) \; \forall i \in I(j_m) \; \forall m \; \forall k$. The order of the active movement groups is free (no cycles are imposed), and the same movement group may be repeated (no constraints on minimum or maximum green times). The structure-free model-based predictive controller follows a rolling horizon approach, repeatedly optimizing a control sequence of movement group states $p_{j_m}(k) \in \{0,1\} \; \forall j_m \forall m$, and corresponding signal states $s_i(k) \in \{0,1\} \; \forall i$, for the upcoming prediction horizon $k = k_0+1, \ldots, k_0+K$, which minimize the total network delay over this prediction horizon, with k_0 referring to the current time interval the traffic system is in and K the number of prediction intervals looked ahead, i.e.,

$$\min_{\{p_{j_m}(k), \forall j_m \, \forall m \, \forall k\}} \sum_i \sum_{k=k_0+1}^{k_0+K} x_i(k) * T \tag{4.1}$$

s.t.

$$s_i(k) = p_{j_m}(k) \quad \forall i \in I(j_m) \; \forall j_m \quad \forall m \; \forall k \tag{4.2}$$

$$\sum_{j_m} p_{j_m}(k) = 1 \quad \forall m \; \forall k \tag{4.3}$$

$$x_i(k) = x_i(k-1) + a_i(k) - d_i(k) \quad \forall i \; \forall k \tag{4.4}$$

$$a_i(k) = \sum_{\bar{\imath}} (d_{\bar{\imath}}(k - n_i) * \pi_{\bar{\imath} i}) \quad \forall i \; \forall k \tag{4.5}$$

$$d_i(k) = \min\left\{ r_i * \left(T - T_i^L(k)\right) * s_i(k), x_i(k-1) + a_i(k), d_i^{\mathrm{MAX}}(k)\right\} \; \forall i \; \forall k \tag{4.6}$$

with:

$x_i(k)$ [veh] the number of queued (and delayed) vehicles per movement i, with $x_i(k_0)$ the current system state.

$a_i(k)$ [veh] the number of arrivals at the queue of movement i (set to the demand d_z at network boundaries z, i.e., $a_i(k) = d_z(k - n_i) * \pi_{zi}$).

$d_i(k)$ [veh]	the number of departures from the queue of movement i ($d_z(k)$ the demand at the network boundaries z).
$\pi_{\tilde{\imath}\hat{\imath}} \in [0,1]$	the turn fraction of the vehicles leaving at movement $\tilde{\imath}$ heading for movement $\hat{\imath}$.
n_i	the number of intervals looking back upstream according to the free-flow travel time, i.e., $n_i = \lceil (L_i/v_i)/T \rceil$,
L_i [m]	the length of the dedicated link of movement i,
v_i [m/s]	the free-flow speed of the dedicated link of movement i.
r_i [veh/s]	the average saturation flow rate at movement i.
$T_i^L(k)$ [s]	the loss time switching from red to green, i.e., $T_i^L(k) = T_L$ if $s_i(k) = 1$ & $s_i(k-1) = 0, T_i^L(k) =$ 0 otherwise.
$d_i^{\mathrm{MAX}}(k)$ [veh]	the maximum outflow due to limited space downstream, i.e., $d_i^{\mathrm{MAX}}(k) = \min_{\hat{\imath}}\{(\frac{(L_{\hat{\imath}}-L_{\hat{\imath}}^Q(k-1))}{L_{\mathrm{VEH}}} - y_{\hat{\imath}}(k))/\pi_{i\hat{\imath}}\}$, with
L_{VEH} [m]	the average space occupied by a single queued vehicle,
$L_{\hat{\imath}}^Q(k)$ [m]	the space occupied by the queue on downstream movement $\hat{\imath}$, i.e., queue length $x_i(k) * L_{\mathrm{VEH}}$, or, more precise, in case of a first order queuing model, the space between the position of the last queued vehicle (tail) and the stop line,
$y_{\hat{\imath}}(k)$ [veh]	the number of driving vehicles already present on downstream movement $\hat{\imath}$, i.e., $y_{\hat{\imath}}(k) = \sum_{n=1}^{n_{\hat{\imath}}} \sum_{\tilde{\imath}} (d_{\tilde{\imath}}(k-n) * \pi_{\tilde{\imath}\hat{\imath}})$

Equations (4.4)-(4.6) represent the prediction model of the controller that express the network delay corresponding to a control sequence. A macroscopic store-and-forward approach is used with vertical queues at the downstream end of each dedicated link of a movement (4.4). Vehicles are assumed to travel with equal free-flow speed to the downstream end of the dedicated link and enter the queue (4.5). Vehicles leave the queue when the signal is green, and if there is enough space at the dedicated links of the downstream movements (4.6). To determine the available space downstream the queues are assumed to be horizontal, and a first order queueing model is applied to determine the occupied space of the queue (for more details see Chapter 3). According to the prediction model, a vehicle is queued and delayed if it arrived at the downstream end of the link according to free-flow conditions but has not departed yet due to a red signal, or queued earlier arrived vehicles, or a blockage downstream.

The delay objective (4.1) is a function of the predicted queue states $x_i(k)$ $\forall i$ $\forall k$ in the network, the decision sequence $p_{j_m}(k)$ $\forall j_m$ $\forall m$ $\forall k$, and the parameters in the prediction model, like the saturation rate r_i, turn fractions $\pi_{\tilde{\imath}\hat{\imath}}$, free-flow speed v_i, and queueing model parameters, and can more generally be expressed as

$$\min_{\{\boldsymbol{p}_k, k=k_0+1,\dots,k_0+K\}} \sum_{k=k_0+1}^{k_0+K} c_k\big(\boldsymbol{x}_k(\boldsymbol{x}_{k-1}, \boldsymbol{p}_k, \boldsymbol{\theta})\big) \tag{4.7}$$

with $\boldsymbol{x}_k$ the vector of queue states $x_i(k)$ $\forall i$, $\boldsymbol{p}_k$ the vector of movement group states $p_{j_m}(k), \forall j_m$ $\forall m$, $\boldsymbol{\theta}$ the vector of all model parameters, and c_k the cost for time index k as a function of the queue state vector $\boldsymbol{x}_k$, i.e., $c_k(\boldsymbol{x}_k) = \sum_i x_i(k) * T$. Note that $\boldsymbol{x}_k$ depends on $\boldsymbol{x}_{k-1}$, $\boldsymbol{p}_k$, and $\boldsymbol{\theta}$ through (4.4)-(4.6).

Robust control

In the optimal control problem (4.1)-(4.6) it is assumed that the initial queue states $x_i(k_0)$ $\forall i$, the demand input $d_z(k)$ at network boundaries z, and the model parameters of the prediction model are known with certainty and the optimal decision sequence $p_{j_m}(k), \forall j_m \; \forall m \; \forall k$ can be found. The structure-free controller can adapt and anticipate to the detailed time-varying arrival pattern $a_i(k)$ $\forall i \; \forall k$, up to intervals of a few seconds. In practice uncertainties will be present in the structure-free model-based predictive control system. Since the fluctuations in the demand pattern are covered in more detail by the high adaptivity of the structure-free controller, uncertainties in the prediction model itself become more dominant and can have a larger effect on control performance than the remaining uncertainties in the demand input and initial state. To protect against possible performance loss due to these parameter uncertainties in the prediction model, robust principles are introduced to incorporate the uncertainties of the model parameter $\boldsymbol{\theta}$ in the controller (4.7).

To this end, let Ω be the set of possible parameter scenario indices $\omega \epsilon \Omega$, with chance of occurrence defined by a known discrete distribution ϕ. Let $\boldsymbol{\theta}_\omega$ be the vector of parameter values and $\phi_\omega = \phi(\boldsymbol{\theta}_\omega)$ the probability of the scenario with index $\omega \epsilon \Omega$. Then the following objectives are defined, replacing (4.7), explicitly considering the possible range of model parameters (addressing aspects 1-3 as introduced in Section 4.3.1):

1. The **non-robust** delay objective for the average parameter value $\overline{\boldsymbol{\theta}} = \sum_\omega \phi_\omega \boldsymbol{\theta}_\omega$, to use as the optimal control **reference** when no uncertainties are considered and no explicit margins are included to anticipate on a possible performance loss in worst-case parameter scenarios, i.e.,

$$\min_{\{\boldsymbol{p}_k, k=k_0+1,\dots,k_0+K\}} \sum_{k=k_0+1}^{k_0+K} c_k \left(\boldsymbol{x}_k(\boldsymbol{x}_{k-1}, \boldsymbol{p}_k, \overline{\boldsymbol{\theta}})\right) \tag{4.8}$$

2A. The **robust** delay objective with a full set of parameter uncertainty scenarios (and sufficient prediction horizon length) to show the potential of robust control to identify and protect against performance loss in worst-case scenarios in a real-time context, i.e.,

$$\min_{\{\boldsymbol{p}_k, k=k_0+1,\dots,k_0+K\}} \max_{\{\omega \epsilon \Omega\}} \sum_{k=k_0+1}^{k_0+K} c_k\left(\boldsymbol{x}_k(\boldsymbol{x}_{k-1}, \boldsymbol{p}_k, \boldsymbol{\theta}_\omega)\right) \tag{4.9}$$

2B. The **heuristic robust** delay objective with a subset of parameter uncertainty scenarios, to analyze the extent to which a heuristic robust approach can still identify and protect against performance loss in worst-case scenarios, considering the (calculation time) limitations in a real-time context, i.e.,

$$\min_{\{\boldsymbol{p}_k, k=k_0+1,\dots,k_0+K\}} \max_{\{\omega \epsilon \overline{\Omega}\}} \sum_{k=k_0+1}^{k_0+K} c_k\left(\boldsymbol{x}_k(\boldsymbol{x}_{k-1}, \boldsymbol{p}_k, \boldsymbol{\theta}_\omega)\right) \tag{4.10}$$

The subset $\overline{\Omega} \subseteq \Omega$ of relevant parameter scenarios is constructed on the fly during the rolling horizon process of the controller by a heuristic approach identifying the most likely worst-case scenarios (see next section).

3. The **average (**or expected mean**)** delay objective to compare to the worst-case optimization (4.9) that may be overprotective in terms of this average control performance, i.e.,

$$\min_{\{\boldsymbol{p}_k, k=k_0+1,\dots,k_0+K\}} \sum_{\omega\epsilon\Omega} \phi_\omega \sum_{k=k_0+1}^{k_0+K} c_k\big(\boldsymbol{x}_k(\boldsymbol{x}_{k-1}, \boldsymbol{p}_k, \boldsymbol{\theta}_\omega)\big) \tag{4.11}$$

Note that a complete discrete distribution of parameter scenarios with chances of occurrence is assumed in (4.8)-(4.11). However, the different objectives can also be defined for a randomly drawn set of scenarios. How to randomly draw a representative set of scenarios such that enough information on the worst-cases is present in the scenario set to let robust programming work properly is outside the scope of this thesis.

Note that the original non-robust optimal controller (4.1)-(4.6) with objective (4.8) is a {0,1} integer programming problem, NP-hard to solve. The number of possible solutions, (|*J*|^|*M*|)^*K*, rapidly grows with the number of movement groups |*J*| per intersection, the number of intersections |*M*|, and the number of prediction intervals *K*. Although challenging for real-time applications, the problem can be solved fast and accurate enough using heuristic approaches. In this chapter a heuristic branch-and-bound search method is used (for more details and convergence properties see Chapter 3). The problem complexity and calculation time of the scenario-based controller with robust objective (4.9) increases further dependent on the number of scenarios |Ω| used in the optimization process. Therefore, a heuristic approach is suggested in this chapter to limit the number of scenarios that needs to be evaluated in the optimization process.

Heuristic robust control

To limit the number of scenarios needed in the robust minmax optimization (4.10), a heuristic approach is used to construct a relevant subset $\bar{\Omega} \subseteq \Omega$ of worst-case scenarios on the fly during the rolling horizon process of the real-time controller. Each time after the decision sequence $\{\boldsymbol{p}_k^{MIN}, k = k_0 + 1, \dots, k_0 + K\}$ is optimized by (4.10), the subset $\bar{\Omega} \subseteq \Omega$ of relevant parameter scenarios is updated. The parameter scenario $\omega^{MAX} \in \Omega$ that performs worst for the optimal decision sequence, i.e., $\max_{\{\omega\epsilon\Omega\}} \sum_{k=k_0+1}^{k_0+K} c_k\left(\boldsymbol{x}_k(\boldsymbol{x}_{k-1}, \boldsymbol{p}_k^{MIN}, \boldsymbol{\theta}_\omega)\right)$, is added to the scenario subset $\bar{\Omega} \subseteq \Omega$, and the oldest parameter scenario that may no longer be needed is removed. The prediction horizon is shifted, and a new decision sequence is optimized by (4.10) with the updated subset $\bar{\Omega} \subseteq \Omega$ of relevant parameter scenarios. In this way, the complete set of scenarios Ω only needs to be evaluated once after the decision update but does not need to be evaluated in full over-and-over again during the time-consuming min-max decision optimization process, which saves calculation time.

The heuristic approach is outlined in more detail in the flowchart of Figure 4.1. The process starts with an arbitrary subset $\bar{\Omega} \subseteq \Omega$ of parameter scenarios of a user-defined size N, dependent on the update frequency and available calculation time in the control application. The subset $\bar{\Omega} \subseteq \Omega$ is adapted on the fly during the rolling horizon process of the controller selecting the relevant worst-case scenarios from the total set of scenarios Ω. The scenario subset is updated at each time k_0 when a new robust solution is calculated (see Figure 4.1). Note that the scenarios $\omega^{(k)}$ in the subset $\bar{\Omega}$ are labelled by their moment k of addition to the subset. The subset $\bar{\Omega} \subseteq \Omega$ is updated as follows. First the new control decision sequence

$\{\boldsymbol{p}_k^{MIN(k_0)}, k = k_0 + 1, \ldots, k_0 + K\}$ is determined for the upcoming prediction horizon by a minmax optimization (4.10) based on the scenarios in the subset $\bar{\Omega} \subseteq \Omega$ at that moment k_0 (Step 1). After this new decision sequence is calculated, the subset $\bar{\Omega} \subseteq \Omega$ of relevant parameter scenarios is adapted. To this end, the parameter scenario $\omega \in \Omega$ that performs worst for the current optimal decision sequence $\{\boldsymbol{p}_k^{MIN(k_0)}, k = k_0 + 1, \ldots, k_0 + K\}$ is determined by $\max_{\{\omega \epsilon \Omega\}} \sum_{k=k_0+1}^{k_0+K} c_k \left(\boldsymbol{x}_k \left(\boldsymbol{x}_{k-1}, \boldsymbol{p}_k^{MIN(k_0)}, \boldsymbol{\theta}_\omega\right)\right)$, and labelled as $\omega^{MAX(k_0)}$ (Step 2). To assure the subset $\bar{\Omega}$ remains of size N and to make way for the new worst-case scenario, the scenario that may no longer be needed is removed from the subset (Step 3). If there is an earlier added duplicate scenario in the subset, i.e., $\exists\, \omega^{MAX(k)} \in \bar{\Omega}$: $\omega^{MAX(k)} = \omega^{MAX(k_0)}$ with $k < k_0$, the duplicate is removed to add the new labelled scenario. In this way, the labels of the scenarios in the subset always indicate the most recent moment the scenario is identified or confirmed as the worst-case scenario. The oldest scenario in the subset, i.e., $\omega^{MAX(k)} \in \bar{\Omega}$ with $k = \min_{\{\omega^{MAX(\hat{k})} \in \bar{\Omega}\}} \hat{k}$, is not confirmed recently to be a worst-case scenario and therefore is most likely no longer needed in the robust optimization. The oldest scenario in the subset is removed to make way for the new worst-case scenario, if there is no duplicate scenario in the subset to remove. The new worst-case scenario $\omega^{MAX(k_0)}$ is added to the subset (step 4). After the subset is updated, the rolling prediction horizon is shifted and a new decision sequence is calculated by a minmax optimization (4.10) based on the updated subset $\bar{\Omega} \subseteq \Omega$ of relevant parameter scenarios, and the process repeats itself.

In the heuristic approach the complete set of scenarios Ω only needs to be evaluated once after the decision update but does not need to be evaluated in full over-and-over again during the time-consuming min-max decision optimization process, which saves calculation time. The choice of the number of scenarios N in the subset in the heuristic approach defines the balance between the robust control objective, i.e., reducing control performance loss in worst-case scenarios, and the calculation time. Note that if N = |Ω| the heuristic is equivalent to the full robust programming approach. Note that if N=1, only one worst-case scenario is considered, and each decision update probably another scenario appears as the worst-case for which the control performance loss needs to be reduced. The more scenarios are used, the less the scenario set will vary, the less fluctuation in the robust decisions, improving the robust control performance. However, for more scenarios more calculation time will be needed in the minmax optimization. If only a limited number of scenarios is used, calculation time is saved in the minmax optimization, however, the robust control decision may alternate due to fluctuation in the subsets of worst-case scenarios. In this chapter, the choice of the number of scenarios N in the subset is experimentally evaluated, to analyze if the heuristic approach works in a highly adaptive system where the decision is updated every few seconds, leading to a solution that over a longer time period is robust to the combined subsets of worst-case scenarios.

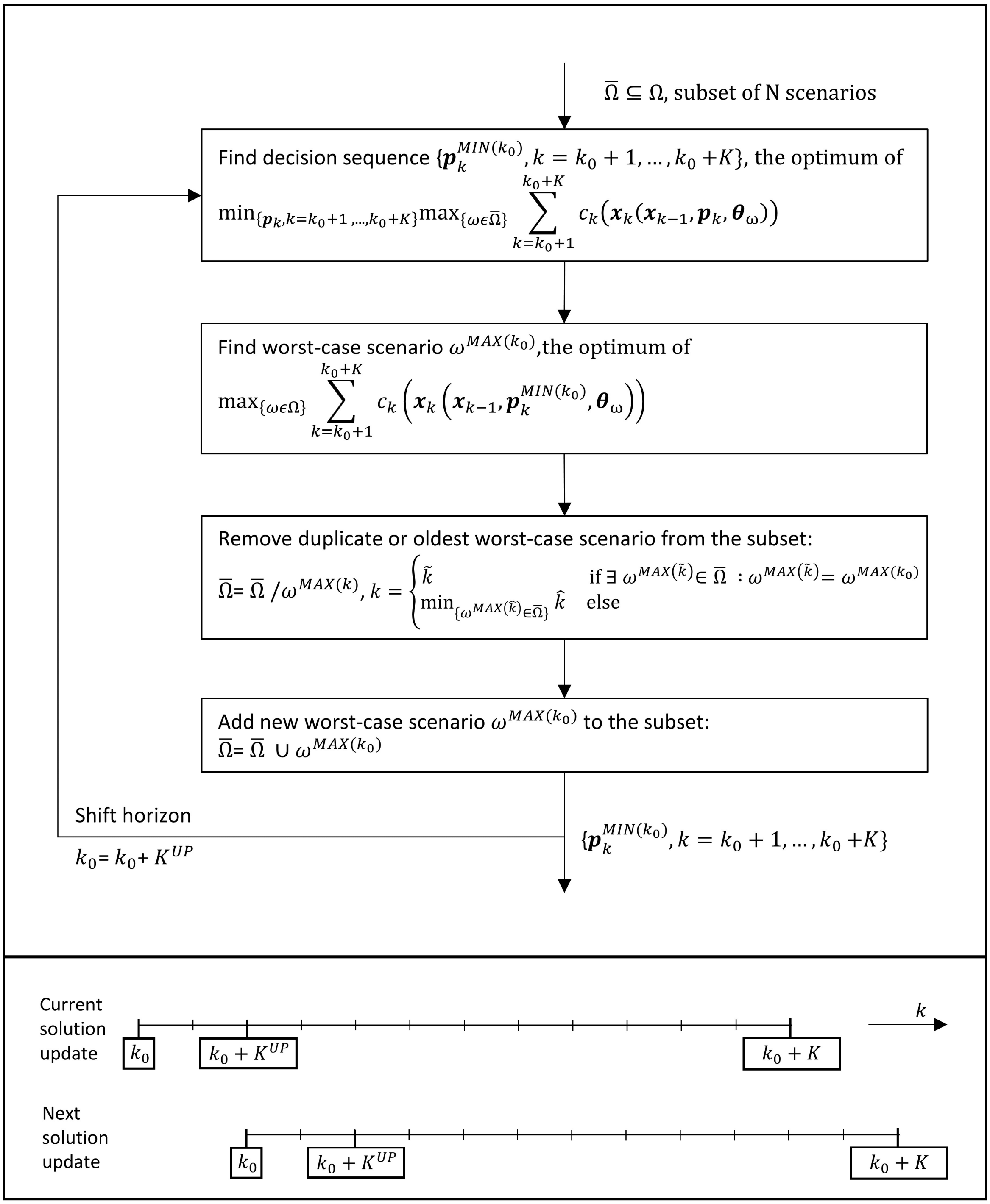


Figure 4.1: Robust solution heuristic.

4.3.3 Simulated traffic case

The robust structure-free model-based predictive controller of Section 4.3.2 is applied in a critical traffic situation, i.e., a corridor with high risk of spillback, to analyze to what extent the robust principles protect in a real-time context against high performance loss due to incorrect model parameters. The traffic case is set up in the microscopic simulator Aimsun [37], with an additional implemented module for the robust control application.

Network configuration

Robust control is designed for critical traffic situations, where prediction errors in model-based predictive control due to model uncertainties can result in an unproportionally large performance loss. In a network with closely related intersections, the effect of prediction errors on the performance of a model-based predictive controller can indeed be unproportionally large causing spillback and blocking traffic at upstream intersections (see Chapter 3). Therefore, the robust controller is applied to a corridor with short intersectional distances and a high change of spillback (see Figure 4.2b). The corridor consists of four intersections, each consisting of the full twelve possible movements considering cars only. There are separate lanes for each movement, only allowing vehicles for this direction. Vehicles choose their destination lane at the intersections, and do not switch lanes in between. Spillback will arise at upstream intersections when the destination lane is full. This is a simplification of reality, but the traffic configuration contains enough spillback effects to generate a break down when the wrong model assumptions are made and to show the potential of robust control.

Demand pattern

Robust control is expected to have most effect around saturated conditions, where there are enough vehicles to cause a substantial effect on the control performance by making a wrong model assumption, but where there is still enough space in the traffic condition to be adjusted by changing the control decision. In undersaturated conditions, the effects of prediction errors due to model uncertainties in model-based predictive control are limited (see Chapter 2) and robust control is not really necessary. In oversaturated conditions, the effects of prediction errors are hard to prevent since there is not enough space in the traffic condition to be adjusted by changing the control decision. Therefore, the robust controller is applied to the corridor with a demand near saturated conditions. The demand pattern [veh/h] consists of a basic level just below saturation, and three oversaturated peaks (see Figure 4.2a), which can be well processed if controlled adequately. The demand is simulated for a period of 25 minutes (with 3 peaks of 1 minute each), with an additional zero demand period to empty the network. Note that the demand is randomly generated according to an exponential distribution of the generation times between vehicles, and the number of realized vehicles differ over time and place at the network boundaries varying around the average demand value. The random seed of the demand pattern is different for the considered parameter scenarios, however, is kept fixed per scenario to compare the different control strategies.

Structure-free controller

Robust model-based predictive control considering model uncertainties becomes valuable if the uncertainties on the parameters in the prediction model are the dominant factor in the control performance, and the fluctuation in the demand is already handled by the adaptivity of the controller. Therefore, the highly adaptive structure-free controller is considered as defined by (4.1)-(4.6), which can easily adapt to the peaks in the demand as defined in Figure 4.2a. At each intersection each movement has its own signal that may change every 6 seconds (T=6), considering a loss time of 3 seconds (T_L=3). The controller has 4 movements group of non-conflicting movements per intersection to choose from (see Figure 4.2c). There is a free choice at an intersection for a specific movement group, which is independent of the choice at the other intersections, and there is no predefined order or imposed cycle in the decision sequence (and no constraint on maximum waiting times or green times). The decision sequence is updated every 12 seconds (K^{UP} =2 intervals) and a prediction horizon of 60 seconds (K=10) is used, large enough to cover most of the dynamics in the corridor. The initial state $x_i(k_0)$ at the movements and demand $d_z(k)$ at the network boundaries are assumed to be adequately known and are retrieved and recorded in advance from the microscopic simulator. This makes the uncertainties in the model parameters $\boldsymbol{\theta}$ the dominant factor in the control performance.

Parameter uncertainty scenarios

Robust control is expected to have most effect for model parameters that are most sensitive, i.e., parameters that influence the control performance most when not predicted correctly. The sensitivity analysis in Chapter 3 shows that prediction errors in turn fractions have one of the largest effects on system performance in a network. Therefore, the robust controller is applied to uncertainties in the turn fractions $\pi_{\hat{\imath}i} \in [0,1]$ of the prediction model (4.4)-(4.6). The other parameters of the prediction model are assumed to be adequately known, and their macroscopic values are set equal to the observed and aggregated values from the microscopic simulation.

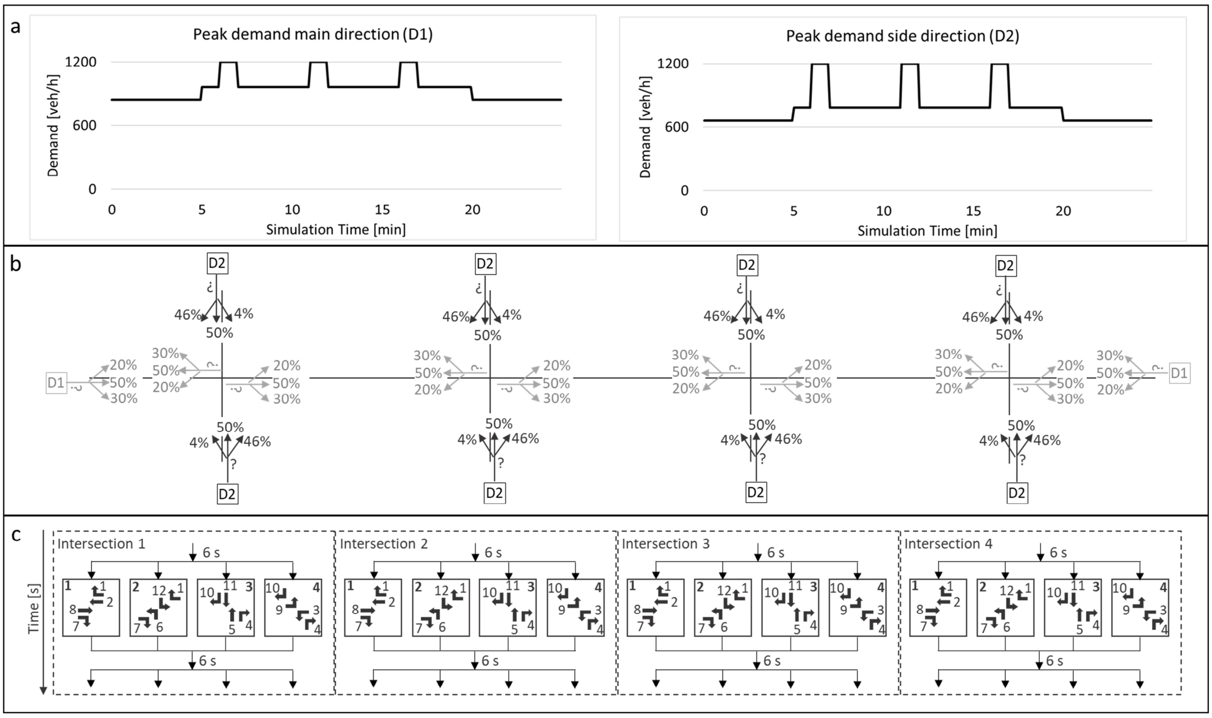


Figure 4.2: Simulated traffic case with (a) demand pattern, (b) network configuration, and (c) structure-free controller.

The uncertainties in the turn fractions are defined as follows. In the basic demand outside the peak periods, the turn fractions are assumed to be known, as well on the main direction of the corridor as from the crossing side direction 50% of the traffic goes in the through direction spreading the traffic over the network as is shown in Figure 4.2b. At the peak periods in the demand, the directions of the vehicles become uncertain, e.g., due to a lack of vehicle information in the suddenly increased demand. The different parameter scenarios $\omega \epsilon \Omega$ are outlined in Table 4.2. The percentage of through traffic in the main direction of the corridor, and the percentage of through traffic from the side direction crossing the corridor are varied independently, resulting in 25 scenarios $\omega \epsilon \Omega$ with different realizations $\boldsymbol{\theta}_\omega$ of the turn fractions. For the chance of occurrence of the percentage of through traffic for main and side direction two independent discrete distributions are defined, resulting in the probabilities $\phi_\omega = \phi(\boldsymbol{\theta}_\omega)$ for the 25 scenarios as given in Table 4.2. In the average parameter scenario, i.e., the scenario with mean parameter values $\overline{\boldsymbol{\theta}} = \sum_\omega \phi_\omega \boldsymbol{\theta}_\omega$, 75% of the total peak traffic (that may be interpreted as all the additional peak traffic) from the side direction, goes in the through direction and crosses the main direction leaving the network. The other 25% of the peak traffic from the side direction turns and enters the corridor. From the peak traffic entering the corridor, 50% enters the main direction, the other 50% leaves the corridor by turning to the side directions. In the other scenarios the percentages of through traffic for main and side direction are changing according to Table 4.2. In the best cases, more peak traffic from the side direction crosses the corridor and leaves the network, and a smaller part of the peak traffic entering the corridor stays on the main direction, alleviating the saturation degree of the corridor. In the worst-case(s), (almost) all peak traffic from the side direction turns on the corridor and (almost) all peak traffic enters the main direction, temporarily increasing the saturation degree of the corridor. If these worst-case turn fractions during the peak period are predicted correctly, the controller can anticipate on the higher saturation degree on the corridor and the corridor can still be controlled adequately. However, when the turn fractions are not predicted and anticipated correctly, the higher saturation degree is not recognized in time, and the risk of spillback increases. Note that the worst-case scenarios have a low probability compared to the average parameter scenario, however, when the worst-case scenarios occur, the performance loss may be relatively high. The robust controller is applied during the peak demand periods to analyze if it can recognize the worst-case scenarios in time and can reduce the loss in performance.

In the experiment all 25 parameter scenarios are simulated. Each of the scenarios is simulated with a different random seed that is kept fixed during the experiment. Note that the random seed does not only affect the demand realization but also the turn fraction realization, letting the turn fractions vary in time, different in each peak, and in place, different at each network boundary and intersection, around the average values, and changing the risk when and where spillback will occur. For all scenarios, the optimal controller (4.7) without uncertainties is applied for the basic demand off-peak periods, in which the turn fractions are known. The different control objectives defined by (4.8)-(4.11) are applied during the peak demand periods with uncertain turn fractions, analyzing (1) the performance loss by the non-robust optimal control decision without parameter uncertainties, showing (2A) the potential of the full robust (minmax) control decision, analyzing (2B) the efficiency of the heuristic robust approach with a limited parameter scenario set, and analyzing (3) the overprotectiveness of robust control by a comparison to the average (expected mean) control decision.

Table 4.2: Turn fraction scenarios with chance of occurrence for peak demand, defined by different through traffic percentages for main and side direction of the corridor.

Parameters $\boldsymbol{\theta}_\omega$: $\omega \in \Omega$	% through on main direction	5%	25%	$\bar{\boldsymbol{\theta}}$:50%	75%	95%
% through from side direction	Probability $\phi_\omega = \phi(\boldsymbol{\theta}_\omega)$	0.05	0.29	0.36	0.18	0.12
95%	0.35	**0.02**	0.10	0.13	0.06	0.04
$\bar{\boldsymbol{\theta}}$:75%	0.38	0.02	0.11	**0.14**	0.07	0.05
50%	0.15	0.01	0.04	0.05	0.03	0.02
25%	0.07	0.00	0.02	0.03	0.01	0.01
5%	0.05	0.00	0.01	0.02	0.01	**0.01**

*Note that the percentage of through traffic is varied, the turn fractions for the right and left directions are adapted proportionally according to the basic turn fractions in Figure 4.2b.
**Note that only traffic arriving and entering the network during the peak period follows the turn fraction scenarios. Once present on the through direction of the corridor, traffic follows the basic turn fractions outlined in Figure 4.2b.

4.4 Results

The overall control performance for the different control objectives, i.e., (1) the non-robust, (2A) full robust, (2B) heuristic robust, and (3) the average control objective, is outlined in Figure 4.3. For the 25 different parameter scenarios, the control performance, expressed in average delay (in vehicle seconds) per minute, is displayed as a function of the simulation time to analyze the performance impact of the peak demands with uncertain turn fractions. The results of three parameter scenarios, i.e., a best-case, the average-case, a worst-case of Table 4.2, are highlighted in Figure 4.3, to compare performance gains and losses in these specific cases for the different control objectives. The total control performance over the entire simulation period is expressed as an output delay distribution over the different turn fraction scenarios weighed by their chance of occurrence (see Table 4.2), to compare the average delays and delay extremes (tail) for the different control objectives. Note that similar statistics are displayed in Figure 4.4, but then for the first peak demand period only, to analyze and zoom in on the performance effect when the different control objectives are truly active and make a difference anticipating on the uncertain turn fractions.

4.4.1 Non-robust control as a reference

The non-robust controller (4.8) with average turn fractions is applied to the 25 scenarios with different turn fraction realizations during the peak demands. Figure 4.3a shows that most of the scenarios can be controlled adequately assuming average turn fractions, increasing delay during the peak demands, and recovering and decreasing delay after the peak demands. These are the turn fractions scenarios that, compared to the average turn fractions, do not increase the saturation degree on the main direction of the corridor, i.e., light colored scenarios in Table 4.2, including the highlighted best and average case.

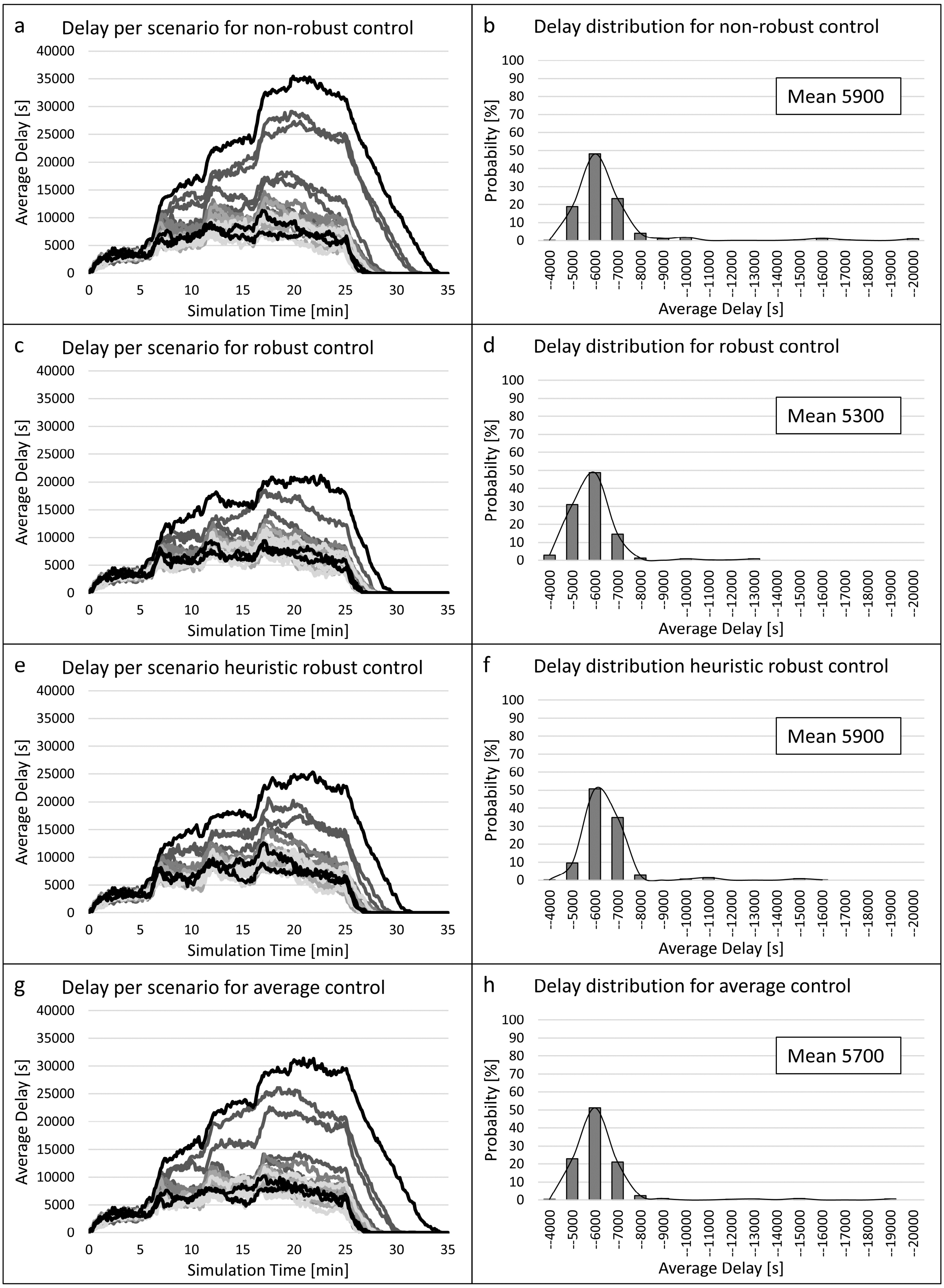


Figure 4.3: Control performance in terms of average delay (vehicle seconds) per minute for all 25 parameter scenarios (3 scenarios highlighted), as a function of the simulation time (left) and as an output distribution (right), for (1) the non-robust, (2A) full robust, (2B) heuristic robust, and (3) the average control objective.

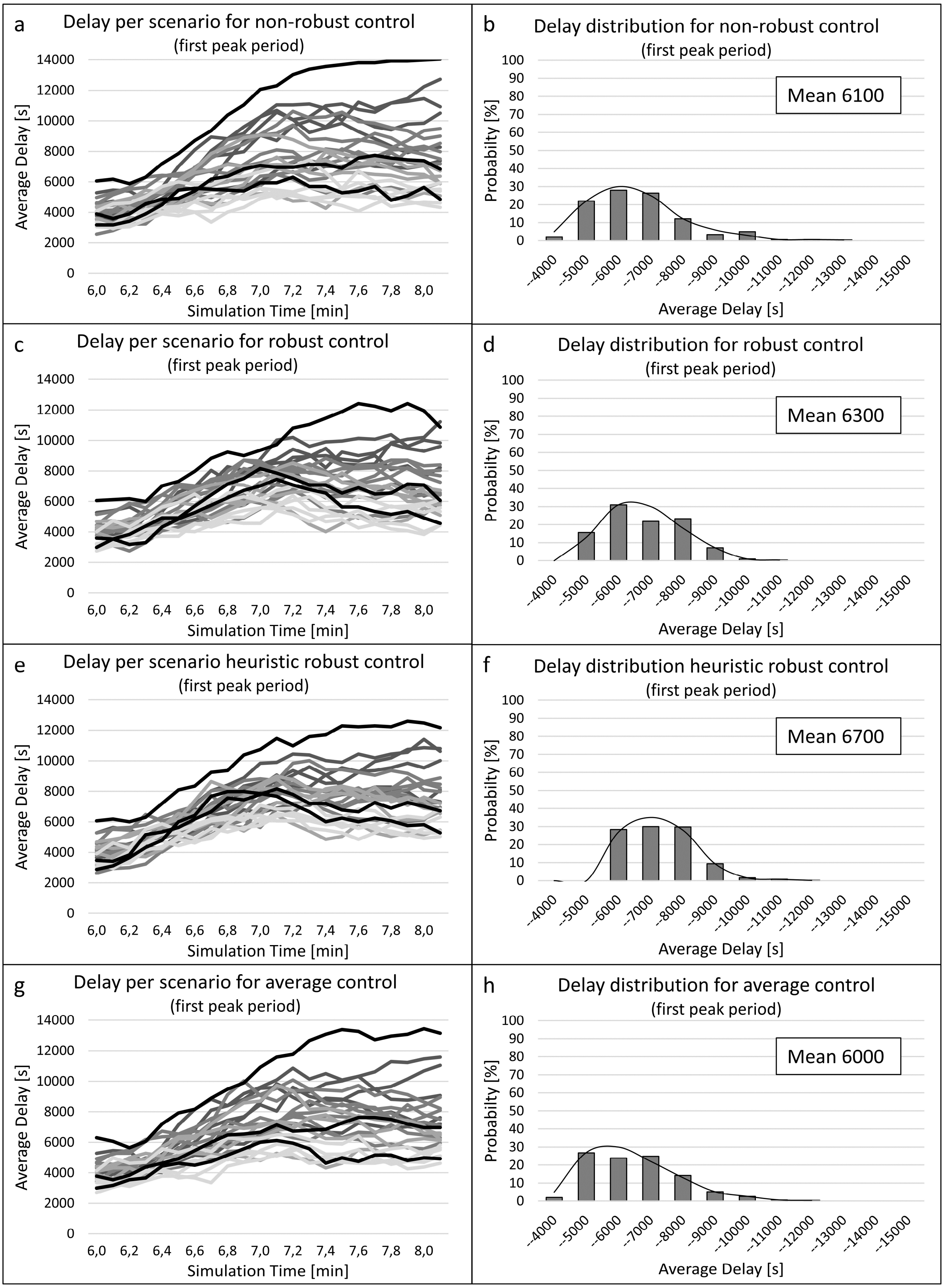


Figure 4.4: Control performance for the first peak period in terms of average delay (vehicle seconds) per minute for all 25 parameter scenarios (3 scenarios highlighted), as a function of the simulation time (left) and as an output distribution (right), for (1) the non-robust, (2A) full robust, (2B) heuristic robust, and (3) the average control objective.

However, some of the scenarios cannot be controlled sufficiently assuming average turn fractions, continuously increasing delay, causing spillback and blocking traffic at upstream intersections. These are the turn fraction scenarios that, compared to the average turn fractions, increase the saturation degree on the corridor, i.e., dark colored scenarios in Table 4.2, including the highlighted worst case. The non-robust predictive controller does not anticipate on a possible increasing saturation degree on the corridor, and although its high adaptivity and fast reinitialization to the actual, now saturated, state the structure-free controller cannot react properly anymore and the damage is already done, resulting in an unproportionally large performance loss (increase in delay). Therefore, the output delay distribution in Figure 4.3b contains a long tail, and although the scenarios with large performance losses do not occur often due to a low chance of occurrence, the mean of the distribution is significantly influenced by these performance losses that are unproportionally large (asymmetric delay). Overall, the real-time predictive non-robust controller that does not consider uncertainties in the parameters at all may result in high performance loss in critical traffic situations due to the wrong assumptions on the parameter values in the prediction model.

4.4.2 Potential of full robust control

The robust controller (4.9) that considers all 25 parameter uncertainty scenarios in the minmax optimization, is applied to all these 25 simulated parameter scenarios with different turn fraction realizations during the peak demands. Figure 4.3c shows that the delay in the worst-case scenarios, i.e., dark colored scenarios in Table 4.2, including the highlighted worst case, is reduced significantly by the robust controller. The robust predictive controller considers all turn fraction scenarios, anticipates on a possible increasing saturation degree in the corridor, and realizes safety margins decreasing the chance on spillback and preventing unproportionally large performance losses. The tail of the output delay distribution (Figure 4.3d) is much shorter than for the non-robust controller. Moreover, the average delay over all scenarios, weighed by the chance of occurrence, is also smaller than for the non-robust controller. The reduction of delay for the worst-cases significantly influences and decreases the mean of the delay distribution (asymmetric delay), although the chances of occurrence are low for these scenarios (Table 4.2). Besides, the robust controller does not have a significant negative effect, rather a slightly positive effect, on the overall performance of the other scenarios, i.e., light colored scenarios in Table 4.2, including the highlighted best and average case. The robust controller is applied only temporarily during the peak demands with uncertain turn fractions (see overprotectiveness in Section 4.4.4), realizing a safety margin in the traffic state, decreasing the saturation degree on the main direction of the corridor, such that when the non-robust controller takes over, the traffic can be controlled more adequately and efficiently with less risk on spillback. Overall, the real-time predictive robust controller that considers all parameter uncertainty scenarios can identify the worst-case scenarios in time and considerably reduces performance loss in critical traffic situations.

4.4.3 Efficiency of robust control heuristic

The heuristic robust controller (4.10) that considers a subset of parameter uncertainty scenarios in the minmax optimization is analyzed for all the 25 simulated parameter scenarios with different turn fraction realizations during the peak demands. Figure 4.3e shows the results for the heuristic with a subset of N=1 scenario. The results for a larger subset of N>1 scenarios lie between the results for N=1 scenario (Figure 4.3e) and the results for the full robust approach of N=25 scenarios (Figure 4.3c). The effect of the size of the scenario subset on the control performance of the heuristic approach is discussed in more detail in the next section. Figure 4.3e shows that the delay in the worst-case scenarios, i.e., dark colored scenarios in Table 4.2, including the highlighted worst case, can still be reduced by the heuristic robust controller, however the reduction is less than for the full robust approach. The heuristic robust predictive controller with N=1 considers alternating worst-case turn fraction scenarios, anticipating and protecting partly to large performance loss due to the possible increasing saturation degree on the main direction of the corridor. The delay in the other scenarios, i.e., light colored scenarios in Table 4.2, including the highlighted best and average case, is also larger for the heuristic approach compared to the full robust approach but also slightly larger compared to the non-robust approach, probably due to the fluctuating decisions. Therefore, the tail of the delay distribution (Figure 4.3f) is longer than for the full robust approach, but still shorter than for the non-robust approach, and also the mean of the delay distribution is larger than for the full robust approach and comparable as for the non-robust approach. Overall, the real-time predictive heuristic robust controller that considers a subset of worst-case parameter scenarios can still considerably reduce performance loss in critical traffic situations, however, may have a negative impact on the control performance under more normal, less-critical traffic conditions due to fluctuating decisions if the subset of worst-case parameter scenarios is too small.

Identification of worst-case scenarios

The choice for the number of scenarios N used in the robust heuristic is analyzed in more detail for the simulations of the three worst-case traffic situations with (5%, 95%), (5%, 75%), (25%, 95%) through traffic (from side, on main) direction (dark colored in Table 4.2), for which the heuristic approach already reduces performance loss considerably with N=1 (see Section 4.4.3). The robust minmax heuristic is applied with an increasing subset size of N=1,2,3,4,5,10 scenarios and compared to the non-robust controller and the full robust controller for these three worst-case traffic situations. The results are averaged over the three worst case traffic simulations to reduce stochastic effects.

The total control performance (delay) as a function of the number of scenarios used in the heuristic minmax optimization is outlined in Figure 4.5a. The delay is indexed to the control performance of the non-robust controller. Note that the results for the non-robust controller are displayed for N=0. Using only 1 scenario, the heuristic robust controller alternates between worst-case scenarios over the subsequent decision sequences, but overall, still reduces the delay in worst-case traffic conditions significantly. Using more, 2 or 3, scenarios, the heuristic approach varies less in the worst-case scenarios in the subset, fluctuates less in subsequent control decisions, and therefore results in a better control performance (lower delay). Using 4 or more scenarios, the heuristic approach only incidentally varies the scenario subset, and gives a comparable performance as the full robust approach with a set of all 25 scenarios. The computation time as a function of the number of scenarios used in the heuristic minmax optimization is outlined in Figure 4.5b. The basic calculation time of the non-robust controller with the average parameter settings and no uncertainty scenarios, is about 2 seconds for the corridor of 4 intersections with 4 movement groups each and a prediction horizon of 1 minute, using a branch-and-bound solution heuristic as proposed in Chapter 3. The computation time of the heuristic robust controller increases linearly with the number of scenarios N used, starting with a calculation time of 2 seconds for N=1 scenario comparable to the non-robust controller, ending up with a calculation time of about 50 seconds for the full robust approach with a set of N=25 scenarios. There is a clear trade-off between calculation time and control performance. Using 1 scenario, the heuristic approach is fastest but loses some control performance. Using all 25 scenarios, the heuristic robust approach guarantees the best control performance but is much slower (too slow for real-time applications if a solution is updated each 12 seconds, K^{UP}=2). Using 3 or 4 scenarios, the heuristic robust approach gives an almost comparable control performance as the full robust approach, but saves computation times considerably, allowing to solve the problem in real time (with a solution update each 12 seconds, K^{UP}=2).

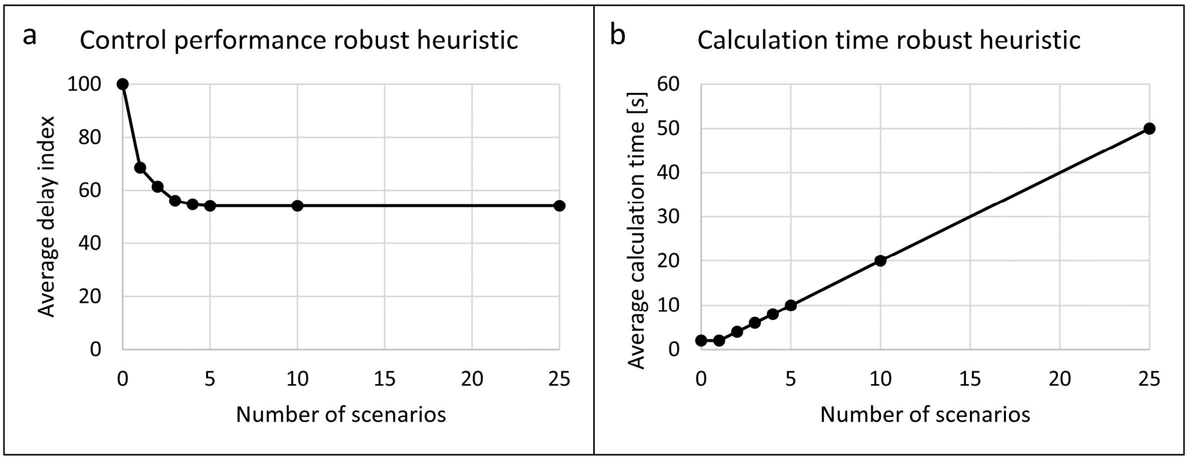


Figure 4.5: Control performance (a) of the heuristic robust approach, indexed to the performance of the non-robust controller, and calculation times (b) of the heuristic robust approach, timed on a computer with an Intel i7-6820HQ processor, as a function of the number of scenarios used in the minmax optimization (0 scenarios represents the non-robust controller). The results are averaged over the simulations for the 3 worst-case scenarios with (5%, 95%), (5%, 75%), (25%, 95%) through traffic (from side, on main) direction.

4.4.4 Overprotectiveness

The robust controller (4.9) is analyzed on overprotectiveness, i.e., a possible decrease in the average control performance over all scenarios as a side effect of building in safety margins to improve the control performance in worst-case scenarios. To study overprotectiveness, the robust worst-case (minmax) delay optimization (4.9) is compared to an optimization of the average (expected mean) delay objective (4.11), both applied to all 25 simulated scenarios with different turn fraction realizations during the peak demands. Figure 4.3g shows that a direct optimization of the average delay results indeed in a reduction of the delay in all the simulated scenarios compared to the non-robust controller (4.8) in Figure 4.3a. The mean of the delay distribution decreases accordingly (see Figure 4.3h compared to Figure 4.3b). Note that the tail of the distribution is only slightly decreased by the expected mean optimization compared to the large decrease in the worst-case optimization (Figure 4.3d), anticipating only partly, with a small chance of occurrence, instead of in full to a possible increase of the saturation degree in the corridor for worst case turn fraction scenarios. The expected mean optimization maintains the traffic situation in the corridor in all simulated scenarios in a more critical state closer to spillback compared to the safety margins that are maintained in the worst-case optimization. Note that the expected mean optimization as well as the worst-case optimization is applied only temporarily during the peak demands with uncertain turn fractions. When the non-robust controller takes over, the traffic can be controlled more adequately and efficiently with less risk on spillback after a robust controlled period instead of after an on average controlled period. Therefore, overall, the average delay for the entire simulation period over all scenarios is lower for the robust minmax controller than for the expected mean optimization (see Figure 4.3d compared to Figure 4.3h).

In first instance the robust controller does not seem to be overprotective, since the average performance (mean of delay distribution) over all scenarios measured for the entire simulation period is not worse but better for the robust controller (Figure 4.3d) than for the non-robust controller (Figure 4.3b), and even better than for the direct optimization of the average delay (Figure 4.3h). However, the robust controller appears to be temporarily overprotective, when analyzing the control performance in more detail for the peak demand periods when the turn fractions are uncertain, and the robust controller is actually applied (see Figure 4.4 for an analysis of the first peak demand period). Figure 4.4 shows that the average delay (mean of delay distribution) over all simulated scenarios during the peak period is higher for the robust controller (see Figure 4.4d) than for the non-robust controller (see Figure 4.4b), indicating overprotectiveness. Furthermore, the direct optimization of the average delay results in the lowest average delay as expected (see Figure 4.4h). The robust controller reduces the delay for the worst-case scenarios the most, but slightly increases the delay of the other scenarios, as shown by the delay lines of the different scenarios that are lying closer together for the robust controller (see Figure 4.4c) compared to the non-robust controller (see Figure 4.4a), decreasing the upper lines, but lifting the lower lines. Accordingly, the delay distribution of the robust controller (Figure 4.4d) is narrower and has the shortest tail, however, is slightly shifted to the right resulting in a higher mean delay value (compared to Figure 4.4b and 4.4h). Overall, the robust controller reduces performance loss in worst-case scenarios considerably, and, consequently, is temporarily overprotective, i.e., the robust controller decreases the average control performance, however only temporarily.

4.5 Conclusion

In this chapter a robust scenario-based minmax optimization is applied to an adaptive structure-free model-based predictive traffic signal controller to protect against performance loss due to uncertainties in model parameters, a dominant factor in these highly adaptive systems. Moreover, a heuristic robust approach is suggested that determines a subset of relevant worst-case scenarios for the minmax optimization on the fly during the rolling horizon process of the controller, which saves calculation time and makes the approach suitable for real-time signal control applications. The (heuristic) robust controller is analyzed for a critical traffic situation, a saturated corridor of intersections with a high chance of spillback during peak periods with uncertain turn fractions. The robust controller is compared to non-robust optimal control to analyze the reduction of performance loss in worst-case scenarios and the robust minmax optimization is compared to an expected-mean optimization to analyze overprotectiveness, i.e., the decrease in average system performance over all scenarios as a side effect.

The analysis shows that the robust controller reduces performance loss considerably in the worst-case parameter (turn fraction) scenarios in the saturated corridor. Non-robust control, based on the average parameter value (average turn fraction), can result in a large performance loss in the saturated corridor, when the wrong assumption is made on the model parameter values (turn fractions), resulting in spillback and blocking traffic at upstream intersections. The real-time predictive controller is too late to correct its decision and the damage is already done, even with the structure-free control freedom and highly adaptive frequent feedback mechanism. Robust control reduces this performance loss considerably, up to 60%, by evaluating all the parameter uncertainty scenarios during the control optimization, identifying and anticipating on the worst-case scenarios in time. Hence, robust control has a large potential in real-time highly adaptive controllers, such as structure-free model-based predictive control systems, to protect against performance loss due to uncertain model parameters.

However, robust control, considering all parameter uncertainty scenarios in the control optimization, is a time-consuming process, for structure-free discrete control decisions in particular, and difficult to optimize in real time. Therefore, a heuristic robust approach is suggested that determines a subset of relevant worst-case scenarios for the minmax optimization on the fly during the rolling horizon process of the controller. The heuristic approach proposed in this chapter reduces the scenario set to 12% of its full size in the robust optimization, saving calculation time while maintaining a similar robust control performance. Note that a robust minmax optimization against only one worst-case scenario that is reconsidered each new decision sequence update, already reduces performance loss in worst-case scenarios to 40%, compared to 60% for the full scenario set, however, decreases the control performance in the other scenarios slightly (5%) due to fluctuating decisions. Increasing the size of the scenario subset to 12% of its original size already removes fluctuations in the control decisions and gives a similar performance in all scenarios as for full robust control. Heuristic approaches, like the one in this chapter, are promising to make robust control suitable for real-time applications.

The analysis in this chapter further shows that the robust controller is overprotective, however only temporarily, decreasing the average system performance only for a brief period in the saturated corridor. Overall, the average system performance over all scenarios is increased in the saturated corridor and is even better than for the direct expected mean optimization. Applying robust control only temporarily in the periods that it is really needed, in this case during uncertain peak periods, a safety margin is applied that leaves the traffic state in a less critical condition in the corridor such that, afterwards, the normal non-robust controller can control the traffic more adequately with less risk on spillback. Identifying worst-cases when the traffic situation will become too critical, and maintaining the traffic state in less critical conditions, is essential for robust control. This can possibly be improved further in adaptive robust control by considering end costs in the model-based predictive controller, which represents the remaining costs that fall outside the scope of the prediction horizon. Besides, a more representative control objective can potentially be used that indicates explicitly the risk on a too critical traffic condition, e.g., the density on the main direction of the corridor that indicates the risk on spillback. In this way, the adaptive robust controller may identify worst-case scenarios better and sooner, protecting only when it is needed (limiting overprotectiveness). This deserves further research.

Overall, the heuristic robust predictive control approach, as proposed in this chapter, provides protection against performance loss due to uncertainties in the prediction model and reduces calculation times, making robust control more suitable for real-time highly adaptive (structure-free) model-based predictive traffic signal control applications.

Chapter 5

Prototyping environment for real-time optimizing predictive traffic signal control applications

For this thesis a prototyping environment has been developed to evaluate predictive traffic signal control applications in a 'controlled' real-time simulation environment to gain insights for the development and design of these predictive control systems in real life. The prototyping environment has been used for sensitivity analysis studying the effect of prediction errors on the control performance of predictive traffic signal control on a single intersection (Chapter 2) and on a corridor of multiple intersections (Chapter 3), and for the development of a robust control approach explicitly considering prediction uncertainties in real-time predictive traffic signal control (Chapter 4). This chapter provides more insight into the prototyping environment itself, such that it can be used for future research and prototyping of real-time predictive traffic signal control applications.

The prototyping environment is available from the GitHub repository:
https://github.com/MurielPoelman/Predictive_Traffic_Signal_Control.git
and a release corresponding to this thesis can be obtained through 4TU.ResearchData:
https://doi.org/10.4121/d19fff77-e5d1-408b-b779-478a7cc88087.v1

5.1 Introduction

For this thesis a prototyping environment has been developed to analyze predictive traffic signal control applications to gain insights for the development and design of these predictive control systems in real life. The prototyping environment is specifically intended for research on model-based predictive control methods, that pro-actively control traffic in a network by repeatedly optimizing control decisions based on a model prediction of the future traffic states. The prototyping environment is set up to study highly adaptive control that reconsiders control decisions frequently and can pro-actively adapt the traffic control signals to match the arrival pattern of individual vehicles. These highly adaptive controllers have a large degree of freedom in the optimization of the control decisions to match the (individual) vehicle patterns, allowing signals to become green in arbitrary order and combination without imposing cycles (structure-free control). The highly adaptive structure-free model-based predictive signal controllers have a large potential increasing control performance but will be challenging and computationally demanding in real-life applications. The prototyping environment is set up to analyze the behavior and potential performance of these kind of advanced predictive traffic signal control systems. The insights obtained in the prototyping environment are intended to be used for the development and design of predictive traffic signal control systems in real life.

The prototyping environment is based on a 'controlled' real-time microscopic simulator for the evaluation of the predictive traffic signal control applications. A real-time microscopic simulation environment is close to the real world, for which the predictive traffic signal control systems are eventually designed. As in real life, the controller can adapt the control decision to the actual traffic pattern distinguishing (simulated) individual vehicle movements through the (simulated) triggering of sensors, and the controller needs to make the control decision in real time. The effect of the control decisions on the overall traffic states (queues and spillback at intersections) is simulated in real time on the scale level of individual vehicles providing detailed insights into the behavior and potential performance of the control system that are representative for real life. Moreover, a simulation is still a ‘controlled’ environment, where experiments can be structurally executed and reproduced. Hence, the designed predictive traffic signal control systems can be structurally studied while functioning close to real life. Therefore, prototyping predictive traffic signal control applications in a real-time simulation environment already shows many of the practical requirements encountered in the design of the controller if applied in real life. These practical requirements ask for solutions in the early development of the control application, which are either additional engineering solutions or adaptations in the underlying scientific control methods. Prototyping predictive traffic signal control applications in a real-time simulation environment makes the design step to the final real-life control application smaller.

For this thesis the prototyping environment has been specifically developed to study and design real-time adaptive structure-free model-based predictive control systems under prediction uncertainties as encountered in real life. A sensitivity analysis has been set up in the prototyping environment to analyze the effect of prediction errors on the behavior and performance of the control system on a single intersection (Chapter 2) and on a corridor of multiple intersections (Chapter 3). A robust control application has been designed in the prototyping environment explicitly considering prediction uncertainties in real-time predictive traffic signal control

(Chapter 4). During the design of the (robust) adaptive structure-free model-based predictive control application in the real-time environment, requirements have been identified as outlined in Table 5.1. These requirements have been addressed and have been implemented in the real-time prototyping environment during the research (Chapters 2-4). Note that the chosen design and implementation is one possible solution and not the solution. Other design choices and implementations may also be able to address the requirements. This chapter describes the prototyping environment given the chosen solution, without considering the other possible design and implementation choices.

Table 5.1: Requirements for adaptive structure-free model-based predictive control applications in a real-time environment. The implementation is described in more detail in the indicated sections.

	Interfacing & Modular Approach	5.2.1 Modules
R1	The traffic signal control application should be an independent application that can be connected to a simulator (similar to the connection to the real world)	5.2.1 Modules Interfacing
R2	The simulator should have an API to connect to an external controller, and a GUI for process monitoring that can run in real time.	Section on Simulator
R3	The control application should have a modular structure to be extendable with different type of controllers (optimizers, prediction models) for prototyping.	Section on Controller
	Parallel Processes	5.2.2 Threads
R4	The control system should be set up as the ongoing process in real life, where multiple parallel processes occur simultaneously and interact with each other (like detecting vehicles, optimizing control decisions, switching signals).	5.2.2 Threads Multi-threaded system
R5	The parallel processes in the controller need to exchange data. Data must be available in time and should not be accessed by different processes at the same time.	Section on Information exchange
R6	The different parallel processes in the controller need to be synchronized to each other and to the simulator (representing the real world).	Section on Synchronization
R7	The parallel processes need to communicate to each other and to the simulator. Although communication times may differ over simulation runs (as in real life), the simulation experiments must be reproducible (for research purposes).	Section on Reproducibility
	Control Methods	5.2.3 Methods
R8	In the ongoing real-time control process, a new decision sequence (control plan) already needs to be calculated while the current control plan is being implemented.	5.2.3 Methods Rolling horizon

R9	In the ongoing real-time control process, a control decision should always be available to implement (even if the control optimization process is not ready yet).	Section on Optimizer
R10	The control optimization method should converge fast, providing a close-to-optimal solution in the limited time available in the real-time control process.*	Section on Optimizer
R11	The control method should be suitable for different control structures, varying from cyclic to structure-free control, and for additional constraints (e.g., max green times).	Section on Control constraints
R12	The control method should be suitable for different user-specified control objectives, that can be expressed in terms of the predicted traffic states.	Section on Control objective
R13	The prediction model should capture the traffic dynamics for nearby intersections and should be fast enough to evaluate candidate control decisions in real time.*	Section on Prediction model
R14	The control method should determine control decisions based on perfect predictions but also based on predictions that are uncertain and contain errors (as in real life).	Section on Uncertainty scenarios
R15	The control optimization method should be able to consider multiple prediction uncertainty scenarios in the limited time available in the real-time control process.*	Section on Uncertainty scenarios

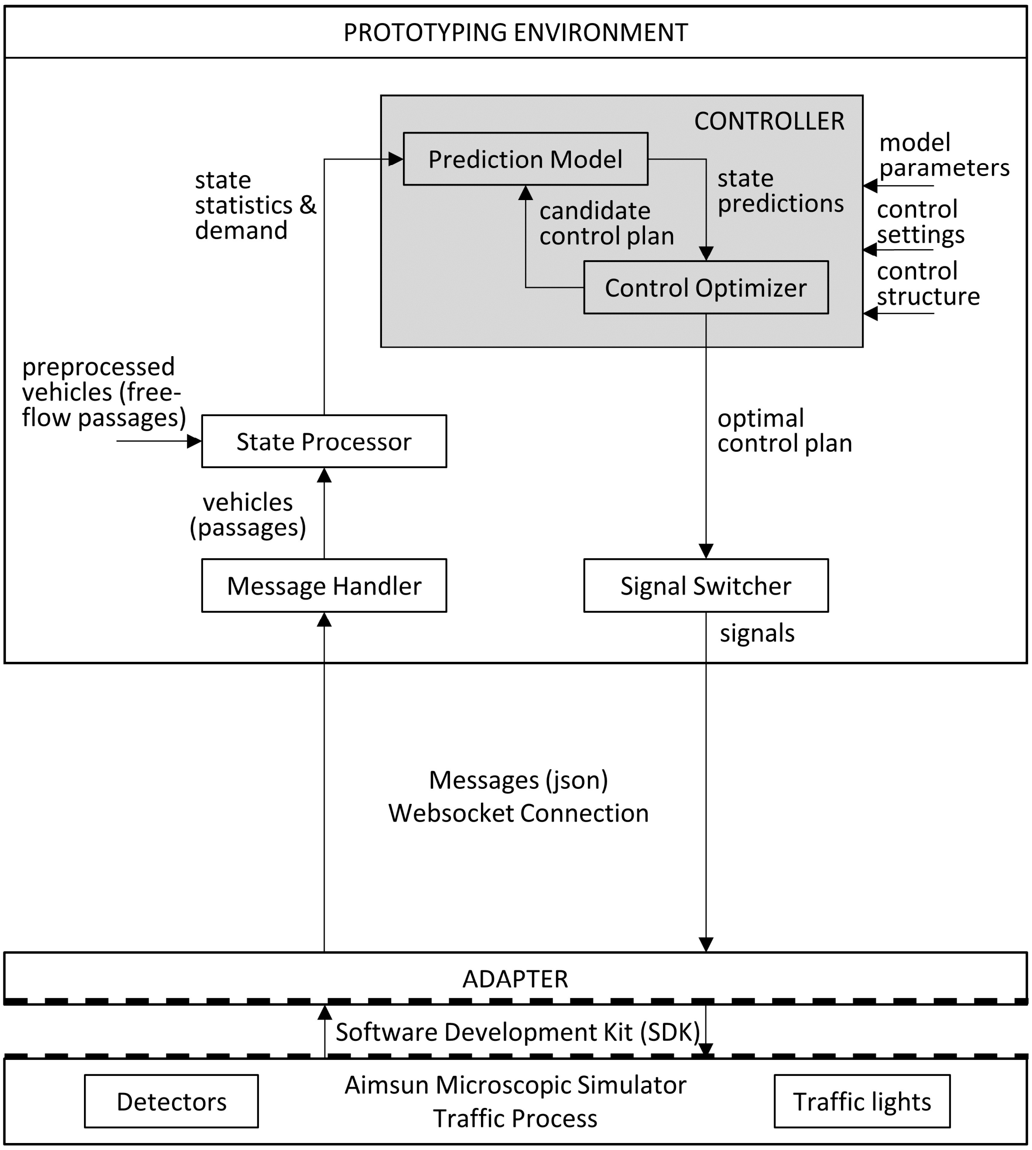
* Requirement is addressed for the problem size of the testcase in this thesis, i.e., a corridor of 4 intersections (Chapters 3, 4).

The real-time prototyping environment has been developed following these requirements, letting the predictive traffic signal control application function as close as possible to its real-life equivalent. However, the prototyping environment is still retained as a controllable environment, in which research experiments can be structurally executed and reproduced. The prototyping environment follows a modular structure, which makes the prototyping environment easy to extend for future research.

5.2 Approach

The functionality and structure of the prototyping environment for real-time predictive traffic signal control is described on multiple levels of detail from general to more specific. The overall application has a modular structure consisting of processes as encountered in real life. The real world is represented by a simulation environment that runs in real time and communicates with a separate traffic signal control application. The control application itself consists of (sub)modules with their own functionality and tasks. The modular structure is described in Section 5.2.1, addressing requirements R1-R3. The modules of the predictive traffic signal control application together form a real-time multi-threaded system of processes that are executed in parallel, just like in real life, i.e., detecting vehicles, processing traffic states, optimizing control decisions, switching traffic signals. These parallel processes all run in real time and need to communicate, share information, and synchronize their actions. The parallel threads are described in Section 5.2.2, addressing requirements R4-R7. The process of optimizing the control decisions based on a prediction of future traffic states, is the main task (and module) of the predictive traffic signal control application. Different methods can be used for optimizing the control decisions, regarding optimization approaches, prediction models, control objectives and constraints, and the incorporation of uncertainties. The available control methods and different options in the prototyping environment are described in Section 5.2.3, addressing requirements R8-R15. Note that in this chapter the functionality and structure of the prototyping environment is described indicating a number of open design choices for the environment. The final design choices are made in the case studies, and are described in detail in Chapters 2,3, and 4.

5.2.1 Modules



[1]Simulator is developed by Aimsun, for more information on this commercial software see https://www.aimsun.com.
[2]Adapter is developed and made available by Path2Mobility & Royal HaskoningDHV, https://www.royalhaskoningdhv.com.
[3]Prototyping environment is developed by M.C. Poelman as open-source software in the context of this thesis, and is available from GitHub repository: https://github.com/MurielPoelman/Predictive_Traffic_Signal_Control.git or a release corresponding to the thesis can be obtained through: 4TU.ResearchData: https://doi.org/10.4121/d19fff77-e5d1-408b-b779-478a7cc88087.v1.

Figure 5.1: Modular structure of the real-time predictive traffic signal control application.

The overall application has a modular structure (see Figure 5.1) and consists of a simulator, an independent control application, and a connecting adapter (R1). A microscopic simulator is used to represent the traffic process in real life. A predictive traffic signal control application is developed as a separate (prototyping) environment that runs in parallel constantly re-evaluating and optimizing control decisions. An adapter (plugged in the simulator) is used to handle the communication between the simulation environment and the traffic signal control application, receiving and sending (json) messages by a WebSocket connection. The total application is originally developed for the microscopic traffic simulation package Aimsun[1] with an adapter designed as Aimsun API plugin[2]. However, other simulation packages can also be used if the adapter is re-designed accordingly. The predictive traffic signal control application is developed as a separate independent prototyping environment[3]. Therefore, the control application can be used in combination with different simulators (as long as the adapter provides similar communication (json) messages). The modular approach makes the prototyping environment generally applicable, following a similar structure and communication process as occurs in real life.

Real-time simulator with adapter

The Aimsun microscopic simulator with adapter (API plugin) runs in real time, simulating a synthetic traffic scenario, visualized in a GUI (R2). The synthetic traffic case, consisting of a network configuration and a traffic demand, needs to be defined in Aimsun (and is passed on to the control application by the adapter at the start of the simulation). The control application is generally applicable to any Aimsun network and demand, however, the application is limited by network size due to real-time calculations. So far, the prototyping environment is set up for a corridor of 4 intersections (with a demand around saturation), which is large enough to include traffic dynamics of spillback, and small enough to guarantee real-time operation. The specific choices on the traffic case depend on the case study performed with the prototyping environment (see traffic scenarios in Chapters 2, 3, and 4). The synthetic traffic case is simulated in real time in Aimsun. The adapter sends messages of Aimsun events, e.g., vehicle entrances, detector passages, or signal changes, to the control application. The adapter receives messages in return from the control application to switch signals of the traffic lights in Aimsun. The traffic evolution and the effect of the adapted control signals can be followed visually in real time in the Aimsun simulation GUI for process monitoring.

Real-time predictive traffic signal controller

The control application runs in parallel to the Aimsun simulator as a separate process (or module). The control application itself also follows a modular structure consisting of subprocesses (or submodules) with their own functionality and tasks (R3). The predictive traffic signal control application re-evaluates and optimizes control decisions in real time following a rolling horizon process. The predictive traffic signal control application receives information on the current traffic state on the level of individual vehicle passages from the simulator (through the adapter). The controller optimizes the control plan, i.e., a control decision sequence for the upcoming time horizon, based on a prediction of future traffic states. A prediction model is used to predict the evolution of traffic states in the network. An optimizer is applied to determine the most effective control plan according to a user-defined objective. The signal states of the optimal control plan are sent to the simulator (through the adapter). The prediction time horizon is shifted, and the process repeats itself. In Figure 5.1 the process is outlined in more detail, consisting of the following subprocesses (submodules):

- Message Handler. The message handler filters the messages of Aimsun events that are sent by the adapter and translates the messages to usable input for the control application. Main input of the control application are the individual vehicle passages at detectors at the stop line of the different movements (signal groups), i.e., individual vehicle passages at green. The message handler stores these individual vehicle passages for later use in the control application.

- State Processor. The traffic state processor aggregates the individual vehicle passages into the current traffic queue states at the dedicated lanes of the different movements (signal groups). Note that additional input is required on predicted free-flow vehicle passages (demand). These free-flow passages are known in the 'controllable' simulation environment and can be recorded in a preprocess simulation under free-flow conditions with green signals and no delays. The traffic state processor combines the pre-processed free-flow vehicle passages with the real delayed vehicle passages and calculates the queues (vehicles that should already have passed under free-flow conditions but have not passed yet). Note that in real life preprocessed free-flow vehicle passages are not available and the demand needs to be predicted, and queue states need to be estimated. To study this subprocess in more detail, the traffic state processor needs to be replaced by a traffic state (and demand) estimator in the prototyping environment.

- Prediction Model. The prediction model evaluates the effect of a candidate control plan for the upcoming time horizon, by making a prediction of the evolution of future traffic states in the network from the current traffic state (and upcoming demand). The prediction model requires additional input that is not provided by Aimsun (through the adapter), regarding model type (general settings) and specific model parameters. For now, a macroscopic store-and-forward model is applied in the control application (see mathematical model formulation in Chapter 3), but any other type of traffic state prediction (model) can be added to the prototyping environment.

- Control Optimizer. The control optimizer determines the most effective control plan (according to a user-defined objective) for the upcoming time horizon. The control optimizer searches the set of possible candidate control plans, repeatedly asks the prediction model to evaluate the effect of candidate control plans in terms of the user-defined objective, and selects the optimal control plan (see Section 5.2.3). The control optimizer requires additional input that is not provided by Aimsun (through the adapter), regarding the optimization search method and objective (general settings) and specific constraints for the possible control plans. For now, a branch-and-bound approach and genetic algorithm are available to search for the control plan with the optimal network delay or total-time-spent (other optimization methods and objectives can be added). The controller is assumed to be structure-free with no imposed cycles or any other constraints in the control plans (see controller in Chapters 3 and 4), but cyclic control with additional constraints can be evaluated in the prototyping environment as well (see controllers in Chapter 2).

- Signal Switcher. The signal switcher activates the optimal control plan. The signal switcher sends messages to the adapter at the right time to switch the signals of the traffic lights in Aimsun.

The effect of the optimal control plans on the traffic process can be analyzed directly in real time in the microscopic simulator (qualitatively in the Aimsun GUI or quantitatively by generating simulation statistics). Additional statistics are generated by the control application to analyze the optimal control plans for the rolling prediction horizons in more detail. The predicted optimal objective value of the optimal control plan (optimizer output) is stored together with the predicted traffic states (prediction model output). The realized objective value for the simulated optimal control plan is stored as well, together with the simulated traffic states (state processor output). A comparison can be made between the predicted and realized control objective and traffic states. In this way, the resulting overall control performance of the model predictive controller can be analyzed, and the accuracy of the underlying predictions can be monitored as well (see analysis in Chapter 3). This additional statistics analysis can be performed outside the control application by using additional software.

The predictive traffic signal control application is designed as a modular environment, consisting of the described subprocesses with their own functionality (R3). The submodules can be replaced, extended, or adapted to the users' need in case studies. The modular approach makes the prototyping environment generally applicable.

5.2.2 Threads

The modules of the real-time predictive traffic signal control application, defined in Figure 5.1 in Section 5.2.1., are in fact separate parallel processes with their own timing that occur simultaneously in real life (R4). Vehicles drive over detectors and generate notifications in parallel, requiring an event-based message handling to process vehicle passages. Signals need to be switched in a separate parallel process at the right times according to the optimal control plan. Statistics need to be collected in parallel as well, at regular time intervals to keep track of the current traffic states in the network, to be ready to use in the control decision process. The actual control decision process, i.e., the control optimization together with the traffic state prediction, is another parallel process that needs to assure there always is an effective control plan available. This process will be executed less often to have enough time to optimize the control plan. During this control decision process, the other processes continue in real time according to their own timing. All processes occur simultaneously and interact with each other.

The real-time predictive traffic signal control application is therefore designed as a real-time multi-threaded system (R4). The threads all run in parallel and communicate, share information, and synchronize their actions in real time. Figure 5.2 outlines the parallel threads and the connections between the threads. The different threads correspond with the different modules defined in Figure 5.1 in Section 5.2.1. Each thread has its own dedicated task and timing (without losing time with side processes):

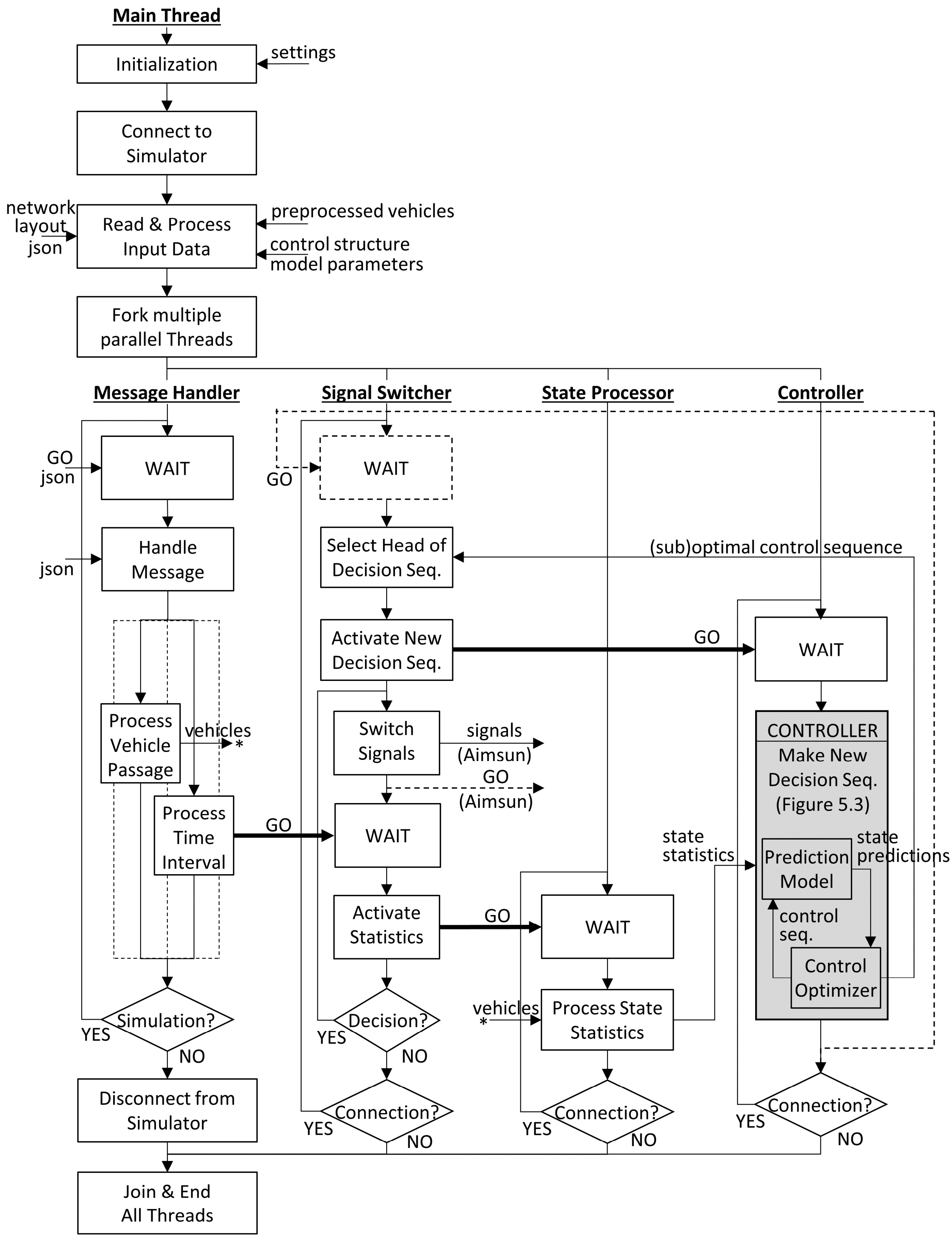


Figure 5.2: Parallel threads in the real-time predictive traffic signal control application, from left to right: thread for the Message Handler, Signal Switcher, State Processor, and Controller (modules from Figure 5.1).

- Message Handler Thread. The Message Handler Thread is the main thread of the predictive traffic signal control application. The Message Handler Thread connects to the simulator, processes the input data, launches the other threads, and handles the messages send by Aimsun (adapter). At the start of the simulation, all information (json) messages are processed regarding the network configuration in Aimsun. The demand is read from an additional file regarding the preprocessed free-flow vehicle passages (used in the State Processor Thread). The user-specified control structure and prediction model parameters are read from files as well (used in the Controller Thread). During the simulation, the information (json) messages from Aimsun are processed (see also Message Handler in Section 5.2.1). The Message Handler Thread waits for a message and handles the message dependent on its kind. If a message on a vehicle passage is received, the vehicle passage is stored for later use in the State Processor Thread. If a message on a fixed elapsed simulation time interval (of a few seconds) is received, the Message Handler Thread orders the Signal Switcher Thread to continue (indicated by "GO" in Figure 5.2) and to switch the signal. Note that in fact the control application uses these fixed elapsed time intervals to synchronize with the simulator (see section on Synchronization). At the end of the simulation, if no more messages are received, the Message Handler Thread disconnects the application from the simulator, ending all threads.

- Signal Switcher Thread. The Signal Switcher Thread receives the optimal control plan from the Controller Thread, consisting of a sequence of control decisions for fixed time intervals (of a few seconds) for the upcoming prediction horizon (of a few minutes). Note that the optimal control plan is not known yet at the start of the simulation and is initialized arbitrary. The Signal Switcher Thread selects and stores the control decisions for the upcoming implementation period, which mostly is only the first part (head) of the total decision sequence for the entire prediction horizon. After the Signal Switcher Thread has received the optimal control plan, the order "GO" is given to the Controller Thread to calculate a new control plan, i.e., control decision sequence for the next rolling prediction horizon. In the meantime, the Signal Switcher Thread implements the current control plan (see also Signal Switcher in Section 5.2.1). The Signal Switcher Thread sends a message to Aimsun (adapter) to switch the signals at the beginning of the next time interval, waits till the time interval has elapsed (an order to continue, i.e., "GO", is given by the Message Handler Thread), gives the intermediate order "GO" to the State Processor Thread to update the statistics for the elapsed time interval, and sends the next message to Aimsun to switch signals, until all decisions of the implementation period are handled and a new control plan is needed. The complete process is repeated as long as the control application is connected to the simulator.

- State Processor Thread. The State Processor Thread waits for an order to continue, i.e., "GO", from the Signal Switcher Thread. Note that this order is sent right after the Signal Switcher Thread has received an order to continue, i.e., "GO", from the Message Handler Thread, when a fixed simulation time interval has elapsed. The State Processor Thread calculates the statistics of the elapsed time interval, by combining the actual delayed vehicle passages (stored by the Message Handler Thread) and the preprocessed free-flow vehicle passages into aggregated traffic queue states (see also State Processor in Section 5.2.1). The current traffic queue states are stored for later use in the Controller Thread. This process is repeated as long as the control application is connected to the simulator.

- Controller Thread. The Controller Thread waits for an order “GO” from the Signal Switcher Thread to start and to calculate a new control plan. The Controller Thread optimizes the control plan, i.e., the control decision sequence for the next upcoming prediction horizon, based on the current traffic queue state (stored by the State Processor Thread) and a model prediction of the evolution of future traffic queue states (see also Control Optimizer and Prediction Model in Section 5.2.1). The Controller Thread stores and updates the best solution found so far, such that there always is a new (sub)optimal control plan available for the Signal Switcher Thread (see section on Information exchange). The process is repeated for the next rolling prediction horizon as long as the control application is connected to the simulator.

The flowchart in Figure 5.2 describes how the multi-threaded application works in real time as an ongoing process. The different threads need to exchange and share information that always needs to be available (R5). The different threads need to synchronize their actions with each other and with the independent simulator (R6). Moreover, the simulation results need to be reproducible since the prototyping environment is meant for research purposes and needs to be a ‘controllable’ environment (R7). These practical, mainly timing related issues are discussed in more detail below.

Information exchange

The different threads need to exchange and share information (R5). Sharing the same data will not be a problem if all threads only use this information as input, leaving the data unchanged. However, shared data can become a problem if data is changed by one thread and used by another thread at the same time, or if the same data is changed by multiple threads. In the control application some data needs to be accessed and changed by multiple threads. For example, the optimal control plan needs to be accessed by the Signal Switcher Thread to activate the control decision, and by the Controller Thread to update the control decision when a next decision or new suboptimal control decision is ready. Therefore, in the control application shared data is temporarily locked when accessed by a thread.

The shared information should always be available, even if the thread that needs to provide this data has not completed its task yet. For example, the Signal Switcher Thread always needs a control plan to implement, even if the optimization process in the Controller Thread may not be finished yet. Therefore, the Controller Thread always provides a reasonable start control decision and continuously updates the best control decision found so far, such that a control plan is always available for implementation (see heuristic optimizer in Section 5.2.3). Hence, shared data is always initialized with reasonable start values, and refreshed as soon as possible with improved values, to assure shared data is always available in the control application.

Synchronization

The traffic process in the Aimsun simulator is the leading process as in real life, and the control application needs to synchronize its actions with the independent simulator (R6). In principle, Aimsun runs in real time using the slowest simulation speed such that one simulation second equals one physical second (timing can slightly deviate). In the control application the time stamps in the (json) messages from Aimsun (adapter) are leading, and the control application synchronizes its actions according to these time stamps. The Message Handler Thread in the control application continuously checks when a fixed time interval has elapsed in Aimsun and passes this information to the Signal Switcher Thread. The Signal Switcher Thread waits on the notification that a fixed time interval has elapsed and orders Aimsun to switch signals exactly at those discrete moments. Note that the synchronization time interval is therefore set equal to the smallest time interval (here, a few seconds) of allowed signal switches in the control application. Moreover, the Signal Switcher Thread also uses the synchronization time intervals to coordinate the actions of the State Processor Thread (aggregating statistics of the passed time interval) and the Controller Thread (making a new decision). In this way, the synchronization assures that the control signals in Aimsun are ordered to switch at the right time as calculated and intended by the control application.

Reproducibility

For research purposes, the simulation results must be reproducible as long as the random number generator seeds are not changed (R7). However, in the designed real-time system no exact reproduction is possible, due to communication lags (analogous to the real-life situation). Even if the control application sends a message to switch signals to the adapter at the right synchronized time, the adapter receives this message slightly later in time and only changes the signals in Aimsun after this time lag. Since the communication lag slightly differs each time that a message is sent, signals can switch a simulation time step (e.g., 0.1 s) sooner or later over different simulation runs, and the simulation results are no longer reproducible. To guarantee reproducibility, Aimsun needs to be stalled before the control application sends the command message and should only continue when the message has been received. In this way, the communication lag is ignored, and signals switch at the same time over different simulation runs, allowing for reproducibility of simulation results. This stop-and-go functionality is made available as an option in the prototyping environment. In this case, Aimsun automatically stops each synchronization interval, and waits for an order to continue, i.e., "GO", from the Signal Switcher Thread, which is sent after the actual order to switch the signals (see dashed output connector to Aimsun in Figure 5.2). Note that in real life the traffic process cannot be stalled. The stop-and-go functionality is available for research purposes only to make simulation results reproducible.

The optimal control decisions for the subsequent rolling horizons should be reproducible as well. However, in the designed real-time system, the moments the optimal control decisions become available cannot exactly be reproduced over simulation runs, due to differences in calculation (processor) times. In the real-time ongoing system, the Signal Switcher Thread cannot wait till the optimal control decision is ready and needs to implement the suboptimal control plan that is available at that moment instead (see also section on Information exchange). Due to differences in the calculation speeds, the new (sub)optimal solution can become available at different times during simulation runs, and therefore, some runs might use the newly optimized plan, and others will use the older plan. To guarantee the use of the same control plans, the prototyping environment contains the option to let the Signal Switcher Thread (and Aimsun as well) wait till the control optimization process in the Controller Thread is finished (see dashed connection between the two threads in Figure 5.2). The Controller Thread can complete the optimization process, to guarantee either optimal solutions or at least reproducible suboptimal solutions by using a convergence criteria or fixed number of iterations. This option not only guarantees reproducibility, but also provides the possibility to compare suboptimal control decisions (calculated in a fixed limited amount of time) to optimal control decisions and analyze the quality of suboptimal control approaches (see heuristic optimizer in Section 5.2.3). Note that in real life the traffic process cannot wait for better control decisions, therefore, this option is meant for research purposes only.

5.2.3 Methods

The main task of the predictive traffic signal control application is the optimization of the control decisions following a model-based predictive control method with rolling horizon. The Controller Thread (see shaded module in Figure 5.2) repeatedly optimizes the decision sequence for the upcoming prediction horizon, based on a model prediction of future traffic states. In a real-time system, a new decision sequence for the next prediction horizon already needs to be calculated while the current control plan is being implemented (R8). The prediction horizon is shifted accordingly. The rolling horizon is illustrated in Figure 5.3a. A control decision sequence consists of control decisions for a sequence of fixed time intervals (indicated by indices k with a length T of a few seconds) for the upcoming prediction horizon (k=k_0+1, …, k_0+K with k_0 the start time and K the number of prediction intervals). The current decision sequence, that is about to be implemented, starts at the current time (k_0 represents "now"). Note that mostly only the first part (head) of the decision sequence is actually implemented (k=k_0+1, …, k_0+K^{UP} with K^{UP} the number of implementation intervals after which the decision is updated), the remaining part is reconsidered in the next decision update. During the implementation of the current decision sequence, the next decision sequence already needs to be optimized, starting at the end of the implementation period (k_0+K^{UP}). The initial state needs to be predicted first for this new starting point (k_0:=k_0+K^{UP}) considering the (head of the) current decision sequence, after which the next decision sequence can be optimized for the shifted prediction horizon (k=k_0+1, ..., k_0+K with k_0:=k_0+K^{UP}).

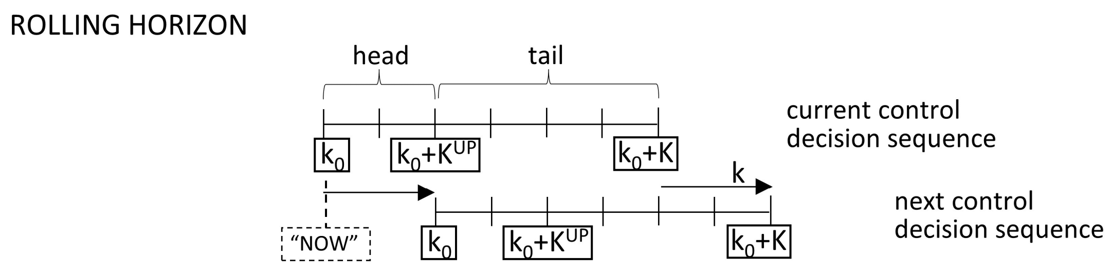


Figure 5.3a: Rolling horizon.

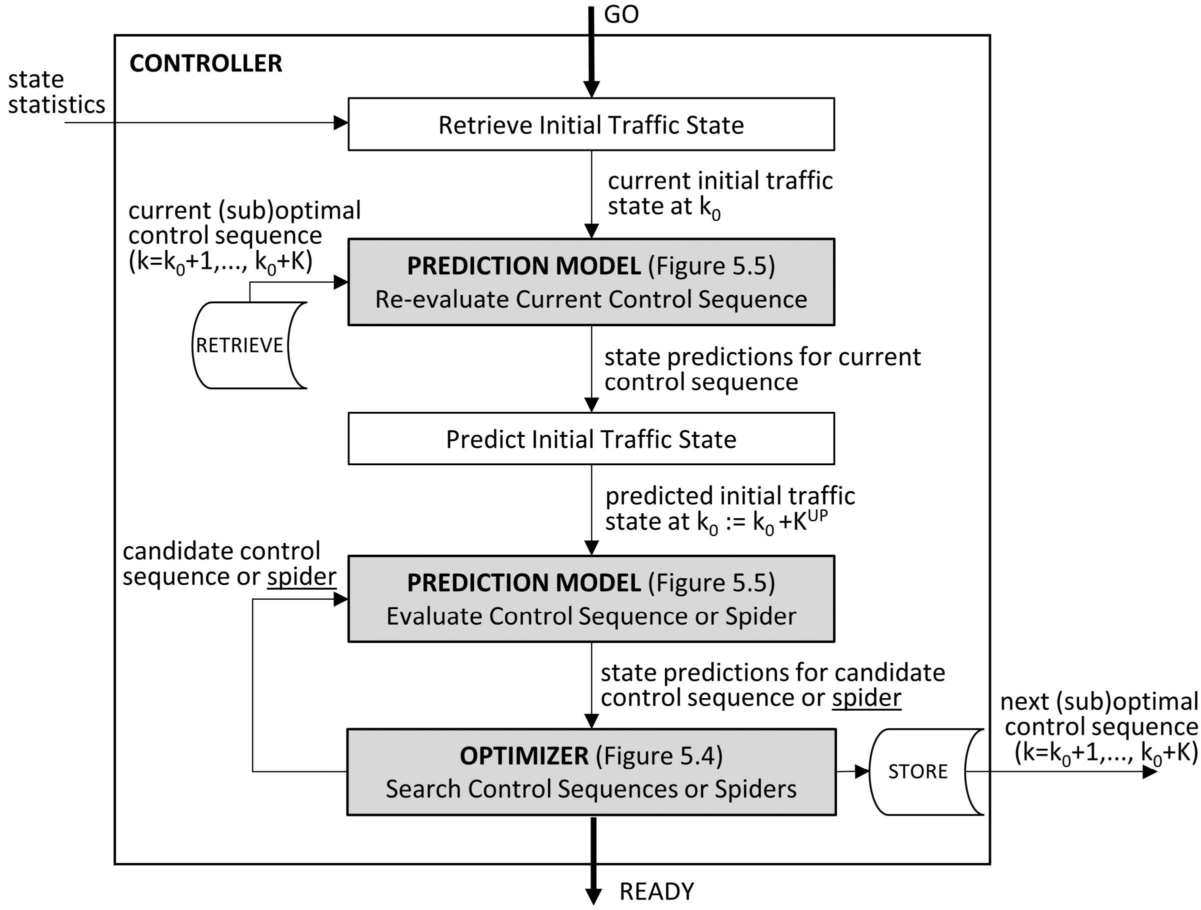


Figure 5.3b: Real-time model-based predictive controller with rolling prediction horizon.

The real-time model-based predictive control method with rolling prediction horizon is outlined in Figure 5.3b. (The outlined controller in Figure 5.3b coincides with the shaded module in the Controller Thread in Figure 5.2.) The current control decision sequence is re-evaluated first for the implementation period, after which the next control decision sequence is optimized for the shifted prediction horizon (R8). To this end, the current initial queue state (and demand) in the network is retrieved from the traffic state statistics (calculated by the State Processor Thread). The current decision sequence is already known and is about to be implemented. A prediction model is used to re-evaluate the known control decisions for the implementation period, making a prediction of the traffic state evolution in the network based on the current initial traffic state. In this way, the traffic state is predicted for the end of the implementation period, providing the (predicted) initial traffic state for the new starting point of the next prediction horizon. The next control decision sequence needs to be optimized for the shifted prediction horizon. A prediction model is used to evaluate candidate control decisions for the upcoming prediction horizon, making a prediction of the traffic state evolution based on the predicted initial traffic state. An optimizer is applied to search through all the possible control decisions, and to find the next optimal control sequence for the upcoming prediction horizon.

Note that different optimizers can be used to optimize the control decisions, and different prediction models can be used to evaluate the candidate control decisions (and to re-evaluate the current control decisions). In the optimization process, different choices can be made (matching the requirements of Table 5.1), regarding the optimization method (R9-10), the control objective and constraints (R11-12), the prediction model (R13), and the incorporation of uncertainties (R14-15). The prototyping environment already contains several options, but users can further extend it. The different methods and possibilities of the prototyping environment are discussed below.

Optimizer

The optimizer determines the most effective control decisions for the upcoming prediction horizon according to a user-defined objective. In the real-time application, only a limited amount of calculation time is available to find an effective control decision (R10), and in the ongoing process, a control decision should always be available to implement (R9).

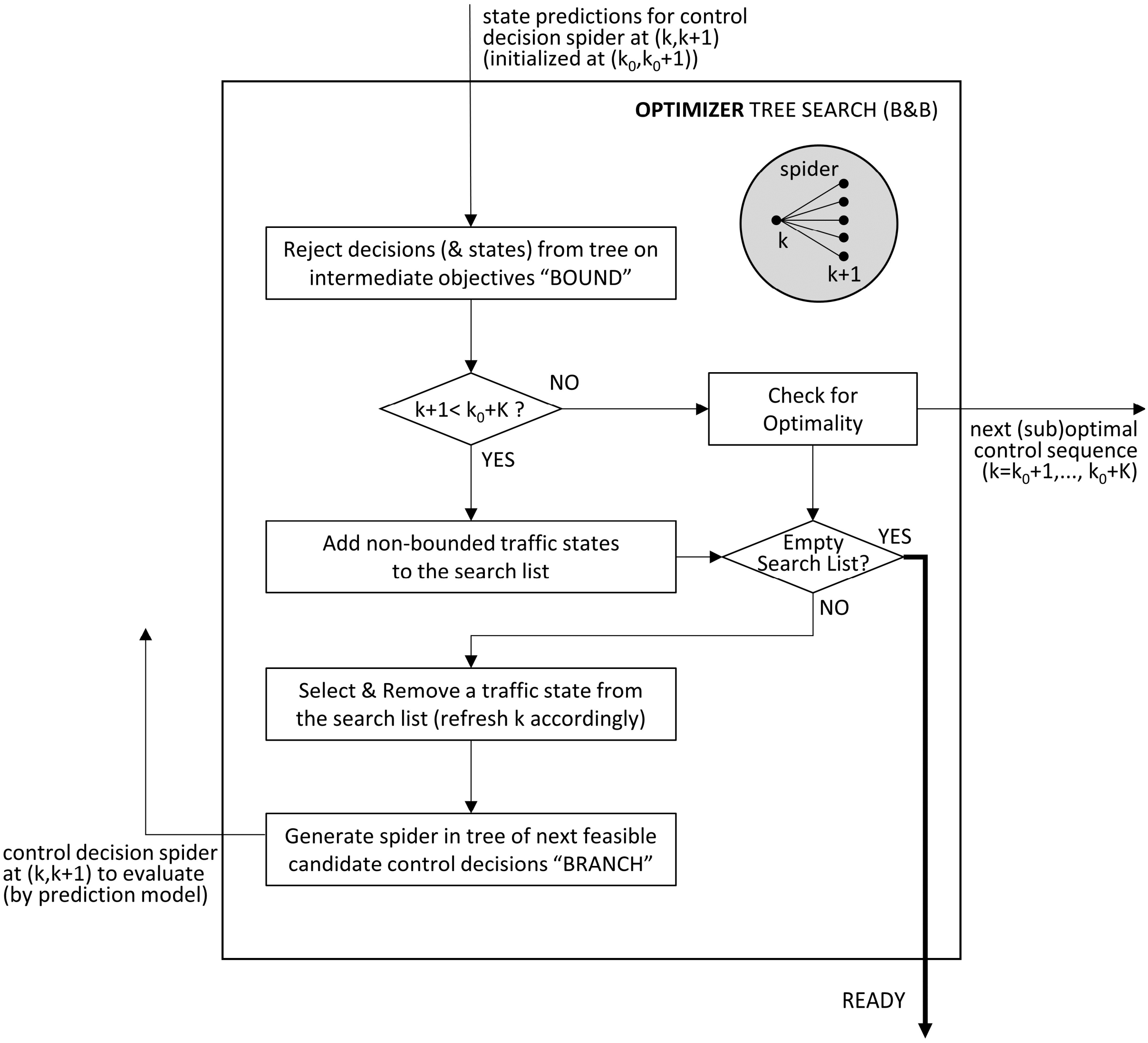


Figure 5.4a: Optimizer based on a tree search method, a branch-and-bound approach.

The optimizer has to search through all the candidate control decisions and needs to determine the control decision with the best objective value, based on an evaluation of the control decisions with a prediction model (see section on Prediction model). Since the exact control decision optimization is a time-consuming search (and evaluation) process, heuristic approaches are needed that converge fast to close-to-optimal solutions in the available calculation time (R10), and that can always provide the best solutions found so far for implementation (R9).

Different kinds of (heuristic) optimizers can be used, with a different approach to search through the candidate control decision sequences. The optimizer can either perform a complete solution search, jumping from one full decision sequence to another (like steepest decent methods or genetic algorithms) or the optimizer can perform a tree-based search, building up decision sequences time interval by time interval (like branch-and-bound methods). Both types of methods are made available in the prototyping environment:

- Branch-and-bound method (tree-based search). A (heuristic) branch-and-bound tree search method is available in the prototyping environment (see Figure 5.4a). The tree search method starts in the initial traffic state, that is branched for all feasible decisions for the next time interval. This initial set (spider) of feasible decisions is evaluated by the prediction model calculating the traffic states and objective values for the next time interval (starting point of the optimizer in Figure 5.4a). The decisions that result in traffic states with too bad intermediate objective values are bound from the tree, not likely resulting in optimal decision sequences. The traffic states with acceptable intermediate objective values are added to a search list to be branched further. If a branch reaches the end of the prediction horizon, the full decision sequence is checked for optimality, and the best decision sequence so far is stored (output of the optimizer). In a repeating process, a traffic state is removed from the search list (often with the best intermediate objective value), the state is branched with the feasible decisions for the next time interval, the set (spider) of candidate decisions is sent to the prediction model to evaluate, and the returned evaluated decisions with calculated states are either bound or added to the search list based on the objective values. The process is repeated until all intermediate traffic states have been processed, and the search list is empty.

  Note that in principle the exact branch-and-bound process can guarantee the optimal solution (if decisions are only bound if the intermediate objective is worse than the full objective of the best complete decision sequence found so far), but then the process can become time-consuming. Heuristic approaches are much faster, already bounding intermediate decisions that do not likely result in optimal decision sequences. Heuristic tree search approaches often use approximations for the tail of the decision sequence (the tail will be reconsidered in the next decision update anyhow), such that the end of the prediction horizon is reached faster, and full decision sequences can be checked more frequently on optimality. Note that heuristic approaches cannot guarantee optimality, and result in suboptimal solutions. In real-time systems, heuristic branch-and-bound approaches are more convenient though, mostly converging faster to complete solutions (R9) often reaching a better solution (R10) than exact methods in the fixed limited available time window. A heuristic branch-and-bound approach with a greedy tail approximation has been developed in this thesis, implemented in the prototyping environment, and analyzed on convergence properties and solution quality (see solution method in Chapter 3).

- Genetic Algorithm (full solution search). As an alternative, a basic genetic algorithm (full solution search method) is available in the prototyping environment as well (see Figure 5.4b). The method starts with an arbitrary set (population) of feasible candidate control decision sequences for the entire prediction horizon. All decision sequences are evaluated one-by-one by the prediction model, calculating the objective values (fitness). The optimal decision sequence is selected and stored as the best solution found so far. If the complete set is evaluated, the set (population) of decision sequences is adapted (as if genetically evolved) by identifying the most promising decision sequences (selection), combining these decision sequences (crossover), and randomly changing some of the decisions in these sequences (mutation). The adapted decision sequences are evaluated again by the prediction model, and the optimal solution is updated in case of better solution candidates. The process is repeated until the objective value of the optimal solution does not change much anymore, and convergence is reached.

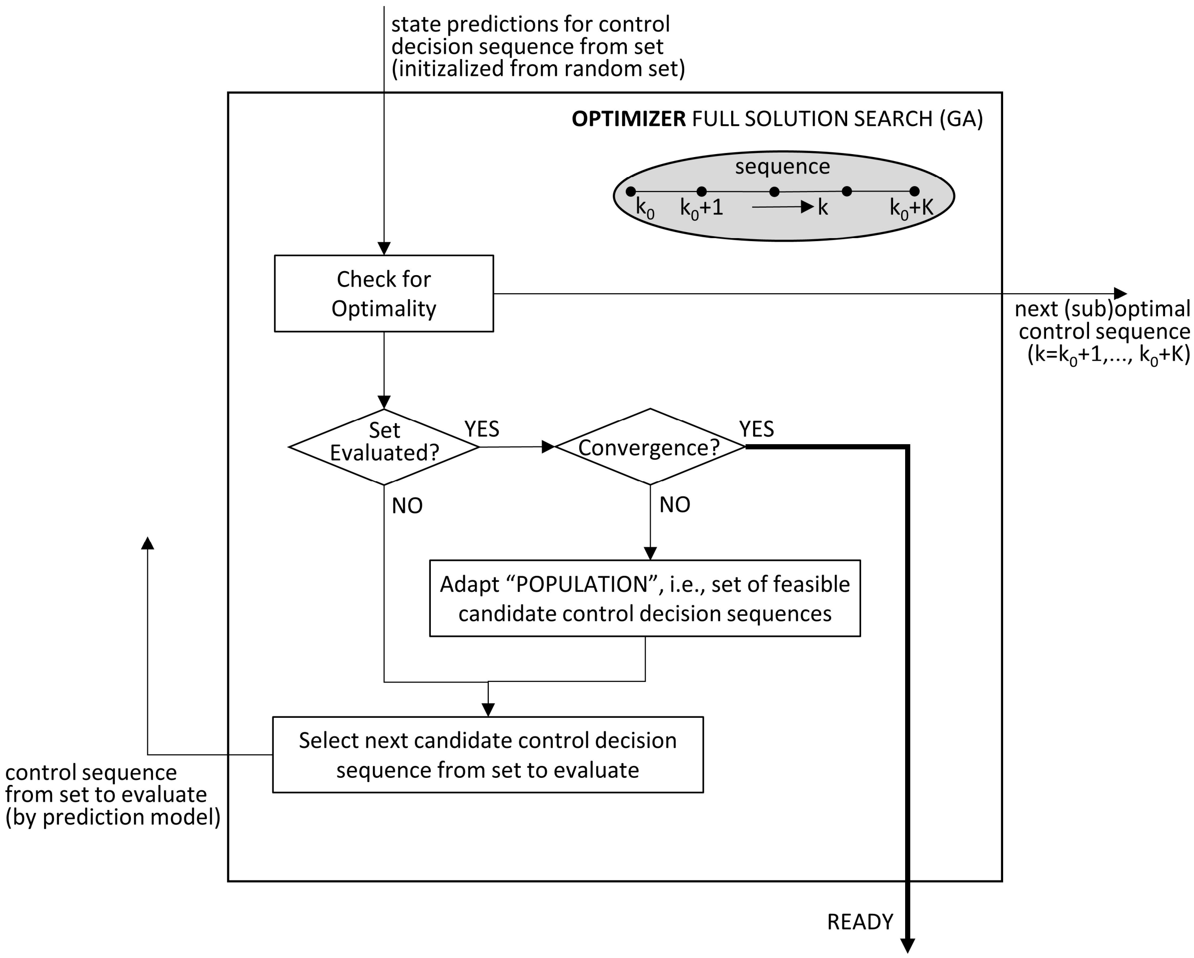


Figure 5.4b: Optimizer based on a full solution search method, a genetic algorithm.

> Note that the genetic algorithm is a heuristic approach by design, converging to a local suboptimal solution, without guaranteeing optimality. In real-time systems, genetic algorithms are convenient, since the full solution search always provides the best solution found so far, ready to be implemented. The genetic algorithm may be a suitable alternative for the heuristic branch-and-bound approach if its basic algorithm is developed further. (Note that in its current form the basic genetic algorithm performs less good and converges slower than the developed branch-and-bound heuristic with greedy tail). This is left for further research.

More (heuristic) optimizers can be added to the prototyping environment (also making use of existing optimization libraries and solvers). Note that the optimizer should converge fast (R10) and should keep track of the best solution found so far (R9), such that a (close to) optimal decision sequence is always ready to be implemented in the real-time control system.

Control constraints and objective

The control optimizer searches through the candidate control decision sequences to find the optimal control decision sequence (see previous section). The candidate control decision sequences need to be feasible, according to user-specified control constraints (R11). In principle, the prototyping environment supports (unconstrained) structure-free control with complete control freedom (see controller in Chapter 3, 4). The signal states of the different movements at the intersections can be activated in arbitrary combination and order without imposing cycles. The only restriction is that the different movements that get green at the same time are non-conflicting. These movement groups need to be predefined for the individual intersections (in control structure files), and the control application automatically considers all combinations of active movement groups over all intersections and time intervals. The prototyping environment offers the possibility to add constraints, excluding candidate control decisions and reducing the degrees of freedom in the control application. A predefined order of movement groups can be specified (in control structure files) and additional constraints on minimum and maximum green times can be defined, such that cyclic control can be studied as a comparison as well (see controller in Chapter 2).

The control optimizer needs to find the most effective control decision sequence according to a user-defined objective (R12). For now, the controller can optimize either delay (vehicle loss hours) or total time spent (vehicle hours) in the prototyping environment (see controllers in Chapter 2, 3, 4). The control application considers the objective as a total network objective calculated over all intersections. The controller optimizes control decisions for all intersections in the network simultaneously. The control decisions of the different intersections are thereby implicitly coordinated, together achieving the optimal user-specified network objective. Any user-specified network control objective can be added, as long as the objective can be calculated based on the outcome of the prediction model (see next section).

Prediction model

A prediction model is used to evaluate the effect of candidate control decision sequences (or spiders) on the total network objective (R13). Note that the prediction model is also used to re-evaluate the current control decision sequence (see Figure 5.3b). The process of the evaluation of control decisions by the prediction model is outlined in Figure 5.5. The evaluation of a decision sequence starts in the initial traffic state. The control decision for the upcoming time interval is activated in the prediction model. The prediction model determines the effect of this control decision on the traffic state looking one time interval ahead, calculating the resulting traffic state at the end of this time interval. The control objective value is determined accordingly. The process is repeated for the next control decision of the next time interval in the sequence until the end of the time horizon is reached, resulting in a sequence of predicted traffic states and corresponding objective values. Note that the evaluation process of a set (spider) of decisions for a tree-search method is similar as the evaluation of a decision sequence for a full solution search method. For a spider the evaluation process starts in an arbitrary intermediate traffic state in the tree, and the decisions in the set (spider) are processed iteratively, repeatedly evaluating the same time interval by the prediction model. The prediction model for one time interval is set up as a separate module in the control application that can be specified by any user-defined traffic state prediction method.

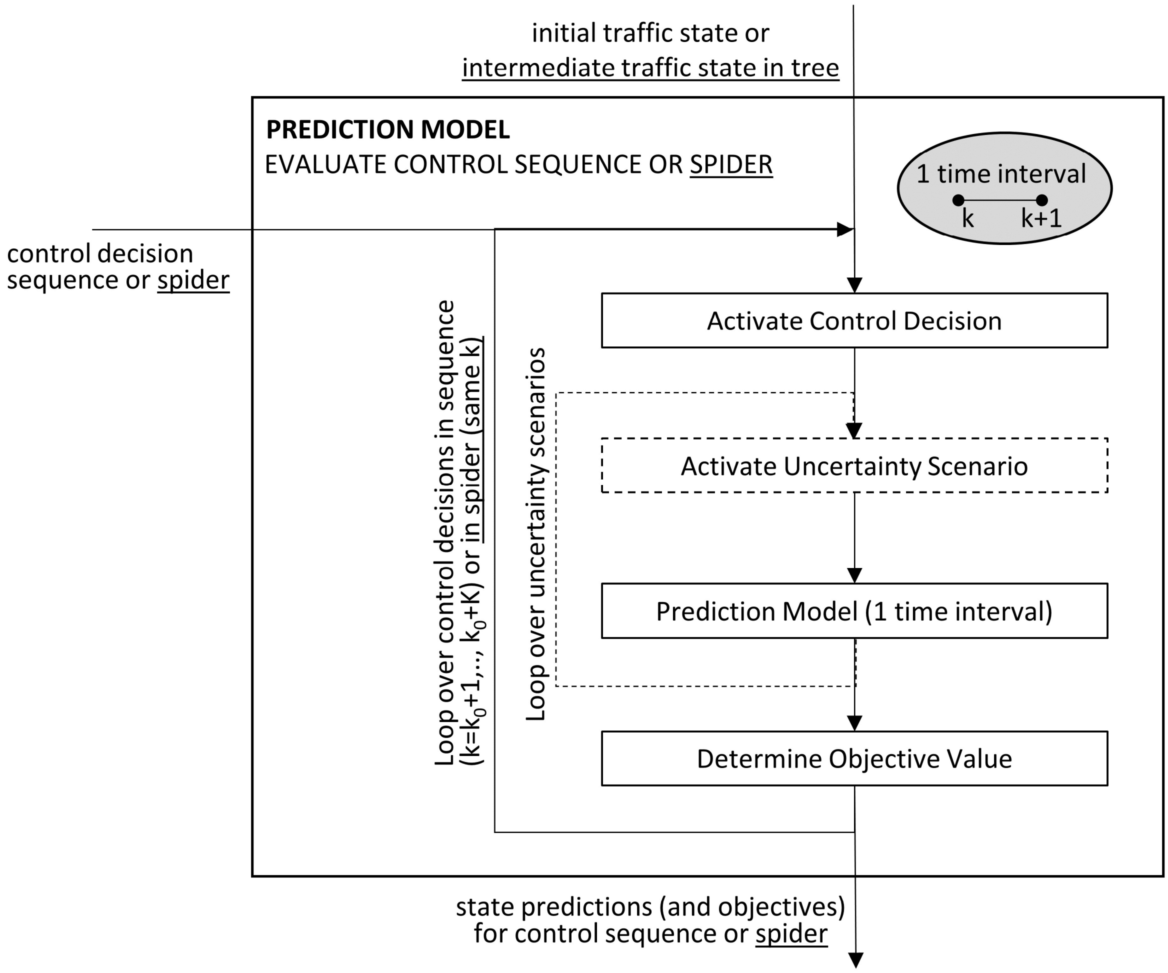


Figure 5.5: Prediction model with uncertainty scenarios to evaluate control decisions.

The prediction model that is available in the prototyping environment is a macroscopic store-and-forward model with first order queueing (see mathematical description in Chapter 3). The parameters of the model are user specified (defined in model parameter file) and, in principle, this parameter setting is fixed in the model evaluation (see section on Uncertainty scenarios). The model describes the input-output relation for an intersection, i.e., vehicle arrivals in the queues and departures from the queues. The model connects intersections by describing the traffic flow propagation in the downstream direction (vehicles traveling between intersections) and in the upstream direction (spillback of queued vehicles). The store-and-forward model with simplified queueing is applied, since the model is accurate enough to describe the traffic dynamics in a small network (with short intersectional distances) and the model is fast enough for real-time applications (R13). Other traffic state prediction methods that are suitable for real-time systems, can be added to the prototyping environment.

Uncertainty scenarios

Different assumptions on prediction uncertainties can be made in the control optimization process (R14), to study ideal control applications with perfect predictions (only possible in a 'controllable' simulation environment) as well as disturbed control applications affected by prediction uncertainties (as appear in real life). In principle, a fixed and well-tuned prediction model parameter setting is assumed in the control optimization process, together with a known state and demand input (directly retrieved from the simulator). In this way, the ideal predictive traffic signal control application can be simulated and analyzed in the prototyping environment, resulting in the best reachable control performance. However, a pre-defined disturbed prediction model setting can be used in the control optimization process instead (together with a disturbed state and demand input). In this way, a disturbed predictive traffic signal control application can be simulated and the effect of the assumed prediction error on the control performance can be analyzed and compared to the ideal control performance (see sensitivity analysis in Chapter 2, 3). Moreover, the control application can explicitly consider multiple uncertainty scenarios of prediction model settings in the optimization process (R15). In this way, robust control applications can be developed and simulated that can reduce the effect of prediction errors on the control performance (see robust control in Chapter 4).

The scenario-based optimization is available as an additional option in the control application (R15). In this case, a set of uncertainty scenarios is generated before a new control decision sequence is optimized. The uncertainty scenarios can either be pre-defined parameter scenarios (read from user-specified files) or can be randomly drawn around the mean parameter values. Note that the pre-defined parameter scenarios are the same for each control decision optimization for all rolling horizons, whereas the randomly drawn scenarios differ in time and place for each new control decision optimization. During the decision optimization process, however, the parameter scenarios are fixed. The uncertainty scenarios are applied each time the prediction model needs to evaluate a control decision (see dashed block in Figure 5.5). The prediction model evaluates the control decision for the upcoming time interval, calculating the effect on the traffic state and corresponding objective value. The decision is evaluated for each uncertainty scenario, repeatedly activating the parameter scenario and applying the model, resulting in different traffic states and objective values for the different scenarios. The objective values are aggregated over all scenarios, and the resulting overall objective value is used in the optimizer. The overall objective in the scenario-based optimization can be defined in different ways, varying from selecting the objective of only one scenario (no uncertainties), to averaging the objective values over all scenarios weighted by the chance of occurrence (expected value optimization), or taking the maximum objective value of the scenarios (robust worst-case optimization). Scenario-based control has been applied in Chapter 4, comparing the different optimization objectives (more objectives can be added to the prototyping environment). Note that the scenario-based optimization increases calculation time because all uncertainty scenarios need to be evaluated. A heuristic approach has been developed in this thesis (see Chapter 4) that needs to evaluate only a limited number of scenarios in the (robust) control optimization, constructing a subset of relevant (worst-case) scenarios on the fly during the rolling horizon process of the controller. The heuristic approach makes the scenario-based optimization more suitable for real-time applications (R15). Scenario-based optimization can be explored and developed further in the prototyping environment.

5.3 Conclusion

A prototyping environment has been developed for the evaluation of predictive traffic signal control applications to gain insights for the development and design of these predictive control systems in real life. The prototyping environment is based on a ‘controlled’ real-time microscopic simulator (Aimsun) to evaluate the control systems. The predictive traffic signal control systems can be studied close to real life, however still by structural experiments in a controllable simulation environment. In the design of the real-time prototyping environment for predictive control, the practical requirements of Table 5.1 are successfully addressed and implemented. Note that the chosen design and implementation is one possible solution and not the solution. Other design choices and implementations may also be able to address the requirements. The requirements of Table 5.1 are implemented in the prototyping environment following a modular approach (R1-R3), considering parallel threads (R4-R7), and making different real-time predictive control methods available (R8-R15) which can be explored and extended in future research.

The prototyping environment follows a modular structure. The control application and Aimsun simulator, representing the real world, are set up as independent parallel processes communicating with each other by sending messages. In this way, the control application can be coupled to other simulation platforms, commercial software packages as well as research open-source simulators (R1). The simulator uses an API adapter plugin managing the messages to communicate with the external controller. The simulator runs in real time, displaying the traffic evolution graphically in a GUI, such that the control process can be monitored as if functioning in real life (R2). The independent control application runs in parallel optimizing the traffic control signals. The control application follows a modular (sub)structure such that different types of predictive controllers can be designed and compared in prototyping (R3). The modular structure makes the prototyping environment easy to extend for future research.

The control application is set up as a real-time multi-threaded system, consisting of multiple parallel processes with dedicated tasks running in parallel as can be distinguished in real life (R4). The parallel processes are synchronized at fixed elapsed simulation time intervals (of a few seconds), adapting to the simulator as the leading process (R6). The parallel processes share common data, which is protected and temporarily locked when accessed, such that data cannot be used by a process while another process modifies the data (R5). Moreover, the common data is always initialized with reasonable start values and refreshed with improved values as soon as possible, such that (reliable) common data is always available (R5). In principle, the control application makes control decisions on the fly, including small communication lags before activating the control decisions in the simulator. However, additional stop-and-go functionality is made available in the prototyping environment to temporarily freeze the ongoing simulation process (not possible in real life), and neglect communication times, for research purposes only to force reproducibility (R7). In this way, the real-time control application can be structurally analyzed as a system of parallel processes close to real life, but also as a fully reproducible experimental environment.

In the prototyping environment different real-time predictive traffic signal control methods can be implemented, analyzed, and compared. The control application follows a rolling horizon approach, to assure an ongoing control process, in which a new control plan is already determined while the current control plan is being implemented (R8). The control application is designed for model-based predictive control, that repeatedly optimizes control decisions based on a model prediction of traffic states. A macro store-and-forward model with simplified queuing is provided as a prediction model, since this model captures the traffic dynamics in a small network and is fast enough for real-time applications (R13). Different traffic state prediction models can be implemented though, or even other more data-driven prediction methods could be considered. Different type of controllers can be analyzed as well with different user-specified control objectives (R12) and control constraints (R11), varying from the more constrained cyclic controllers adapting to aggregated traffic flows, to the highly adaptive structure-free controllers without imposed cycles matching individual vehicle patterns.

Different optimizers can be studied to find the best control decisions, with a focus on heuristic approaches. A heuristic tree search method, i.e., a branch-and-bound method with greedy tail approximation, has been developed in the prototyping environment, and a heuristic full solution search method, i.e., a genetic algorithm, is available as an alternative. The heuristic branch-and-bound approach converges fast to close-to-optimal solutions in the available calculation time (R10), and always provides the best suboptimal solution found so far ready to be implemented in the ongoing control process (R9). The genetic algorithm has the same potential but need to be developed further. Other heuristic optimization approaches can be added to the prototyping environment. Note that the stop-and-go functionality that temporarily freezes the simulation process makes it also possible to study more time-consuming exact optimization methods as a comparison.

In principle, the different control methods in the prototyping environment are designed under assumptions of perfect predictions showing the potential of the predictive control system (only possible in a simulation environment). However, prediction uncertainties can be included in the prototyping environment as well, to not only evaluate the control system under perfect predictions but also under erroneous predictions as encountered in real life (R14). Moreover, a heuristic robust (scenario-based) optimization method has been developed to explicitly consider uncertainty scenarios in the real-time control process to reduce the effects of prediction uncertainties (R15). New types of real-time predictive traffic signal control methods, with or without uncertainty modeling, can be added to the prototyping environment for future research.

The developed predictive traffic signal control application successfully meets the requirements for real-time prototyping, which makes the design step to real-life predictive control systems smaller. The developed real-time prototyping environment shows the potential of advanced real-time predictive traffic signal control applications as applied in real life. Different aspects of predictive control systems can be studied in this prototyping environment, like the main analysis in this thesis on the behavior of the control system under prediction uncertainties and the development of robust (scenario-based) control approaches. More aspects of predictive traffic signal control can be analyzed in this prototyping environment in future research. The prototyping environment thereby contributes to the development and design of real-time predictive traffic signal control systems in real life.

Chapter 6

Conclusion

In this thesis highly adaptive (structure-free) model-based predictive traffic signal control systems are studied considering prediction uncertainties. According to the main objective of this thesis: (1) A sensitivity analysis framework is developed, and new insights for control design are obtained regarding the influence of prediction errors on the behavior and performance of the predictive control system. (2) These insights are used to design a heuristic robust control approach that explicitly considers uncertainties in the prediction model to reduce the impact of prediction errors on the control performance. In a real-time prototyping environment, specifically set up for this thesis, the performance loss due to prediction uncertainties is analyzed, and the capability of the robust control approach is shown to reduce this performance loss. The robust control approach leads, as aimed, to a predictive traffic signal control system that suffers less from prediction uncertainties, improving the performance of the predictive traffic signal control system under uncertainties.

This thesis shows that predictions have a large potential for improving the performance of traffic signal control systems, if prediction uncertainties are explicitly considered and properly managed. For the highly adaptive (structure-free) model-based predictive traffic signal controller, studied under perfect predictions, a longer prediction horizon leads to an increase in control performance up to a certain optimal prediction horizon length. Network control outperforms local control, connecting multiple intersections by predictive information. The predictive property of the control system remains preserved under erroneous predictions, since enough remaining structure is left in the prediction model to rely on. Especially in a saturated network, connecting intersections by predictive information, even when it contains errors, is essential to control traffic efficiently. The high adaptivity of the control system, i.e., a high update frequency to reinitialize often to the current state by the feedback mechanism and a large control freedom (preferably structure-free control) to correct mistakes quickly, is important to limit the impact of prediction errors. However, in critical traffic conditions, a significant performance loss remains due to prediction errors caused by input errors, aggregation errors, and biases in the prediction model, especially for the most sensitive quantities, i.e., saturation rates and turn fractions. So, more accurate predictions for these most sensitive quantities, and the use of disaggregated individual vehicle data, would improve control performance in future control applications. However, these quantities may also be the most difficult to predict in practice, since they contain the largest uncertainties. Uncertainties in these prediction model parameters should therefore be considered explicitly in the control design. This thesis shows that robust control approaches can be made suitable for real-time adaptive (structure-free) model-based predictive control and reduce the performance loss due to uncertainties in the prediction model parameters considerably. The results in this thesis contribute to predictive traffic signal control systems that suffer less from prediction uncertainties, which increases the potential use of these predictive traffic signal control systems for real-life applications.

This chapter discusses the significance and main finding of this thesis in more detail. In Section 6.1 the key findings are described answering the research questions. In Section 6.2 the contributions of this thesis are summarized. In Section 6.3 the practical implications of the research are discussed, and in Section 6.4 recommendations are made for further research.

6.1 Key findings

6.1.1 Prediction quality

In Chapter 2 and 3 the relation between prediction errors and control performance of highly adaptive (structure-free) model-based predictive traffic signal control systems is analyzed at a local intersection and in a network, as asked for in research questions 1A and 1B respectively. In both chapters, a sensitivity analysis is performed in the developed real-time prototyping environment as described in Chapter 5. In a microscopic simulator the perfect world with perfect predictions is created as a reference, in which the predictive traffic signal controller can reach its potential performance. Prediction errors are structurally introduced one-by-one in this controlled simulation environment, and the effects on the control performance are measured. The sensitivity analysis is performed for local control on a single intersection in Chapter 2, and for network control on a corridor of multiple intersections in Chapter 3. In both chapters the sensitivity analysis results are translated into guidelines for the design of predictive traffic signal control under uncertainties. The sensitivity analysis results show that:

1A *Prediction errors in the input and output of a single intersection decrease control performance significantly for highly adaptive (structure-free) model-based predictive traffic signal control systems on a local-controlled intersection. However, the erroneous input and output predictions are still beneficial for the control system, i.e., an increase of the prediction horizon still improves the control performance although the prediction contains errors.* [Chapter 2]

Under perfect predictions, the performance of the highly adaptive (structure-free) model-based predictive traffic signal control system on a single intersection improves if the prediction horizon is increased. The performance increase becomes less for the longer prediction horizons, until the performance does not significantly improve anymore, identifying the optimal prediction horizon length (see Figure 2.5). Increasing the prediction horizon still improves the performance of the predictive traffic signal control system, in most of the cases with biases in the predicted input, i.e., predicted arrivals and initial queue state, and output, i.e., predicted departures (see Figure 2.6). Increasing the prediction horizon partly reduces the effect of prediction errors in a quantity, by compensating with predictions of the other undisturbed quantities. However, increasing the prediction horizon does not completely reduce the effect of prediction errors, and a significant performance loss remains due to erroneous input and output predictions. The saturation rate, i.e., predicted departures, is the most sensitive quantity, because the saturation rate is a stand-alone quantity for which prediction errors cannot be compensated by information on other quantities (see Figure 2.6). Furthermore, under perfect predictions, the highly adaptive structure-free predictive controller outperforms the more traditional cyclic predictive controller, matching individual vehicle arrival patterns instead of aggregated traffic flows (see Figure 2.5). The prediction horizon length can be set to a smaller value for the highly adaptive predictive controllers with more degrees of control freedom, obtaining a comparable or even better performance than for cyclic predictive control that needs a much longer prediction horizon. Structure-free predictive control, despite its dependency on more detailed predictions of individual vehicle patterns, is not more sensitive for prediction errors than cyclic control that needs less information on aggregated traffic flow predictions. Structure-free predictive control outperforms cyclic predictive control, even under erroneous predictions, because the larger control freedom can correct more mistakes faster (see Figure 2.6). This answers research question 1A and shows that erroneous predictions still have added value in highly adaptive traffic signal control systems on a single intersection, however significant control performance is lost due to input and output prediction errors and can be gained with more accurate predictions especially for stand-alone quantities.

1B *Prediction errors propagating in a network decrease control performance significantly for highly adaptive (structure-free) model-based predictive traffic signal control systems on a network-controlled corridor. However, the erroneous predictions are still beneficial for the control system, i.e., an increase of the prediction horizon, looking ahead over intersections, improves the control performance although the prediction contains errors.* [Chapter 3]

Under perfect predictions, the performance of the highly adaptive (structure-free) model-based predictive traffic signal control system on the corridor improves for an increasing prediction horizon, up to a certain optimal prediction horizon length, after which the performance does not improve anymore (similarly to local control in 1A). Network control outperforms local intersection control. Connecting information on arriving vehicles and spillback of multiple intersections, by increasing the prediction horizon, increases the control performance significantly (up to 75%) in the saturated corridor (see Figure 3.6a). The optimal prediction horizon, typically network dependent, does not need to cover all intersections, as the performance gain increases but flattens out for additional intersections. Longer prediction horizons still improve the performance of the predictive controller, even when prediction errors are propagating in the corridor, originated by errors in the initial state and demand input, or by aggregation errors and additional biases in the prediction model (see Figure 3.6c). The predictive property of the control system remains preserved under erroneous predictions, since enough remaining structure is left in the aggregated and biased prediction model to rely on (as noted for local control in 1A). In a saturated network, connecting intersections by predictions, despite possible errors, is essential to control traffic efficiently. Furthermore, a high adaptivity of the predictive control system, i.e., a frequent control update and a large control freedom, is essential to identify and correct mistakes quickly to limit the impact of prediction errors on the control performance. A higher update frequency, reinitializing to the current state more often, significantly reduces the performance loss (up to 30%) due to prediction errors (see Figure 3.6d). This reduction strongly depends on the quality of the actual state, which is therefore most important to estimate accurately. Although partly reduced, a significant performance loss remains due to aggregation errors (of 20%) and additional biases (of another 25% or more) in the predictive control system (see Table 3.2). In the saturated corridor, aggregation errors and biases in the saturation rate (vehicle departures), as already identified for local control in 1A, and in the turn fractions (vehicle directions) cause the largest performance loss (see Table 3.2). Besides, for all model quantities one direction of the bias appears to have more impact on the system performance than the other direction, indicating the safe side to make predictions in case of large uncertainties (see Table 3.2). This answers research question 1B and shows that erroneous predictions still have added value in highly adaptive traffic signal control systems in a network, however significant control performance is lost due to propagating prediction errors and can be gained with more accurate predictions, especially by including disaggregated information on individual vehicles.

6.1.2 Control robustness

In Chapter 4 robust control approaches are studied for real-time highly adaptive (structure-free) model-based predictive traffic signal control systems on the ability to reduce the effect of prediction uncertainties on the control performance, as asked for in research question 2. Robust worst-case (minmax) scenario-based optimization techniques, originally developed for offline fixed-time control to protect against performance loss due to demand fluctuations, are translated to the new online real-time highly adaptive context of (structure-free) model-based predictive traffic signal control to protect against performance loss due to uncertainties in the prediction model parameters. A new heuristic robust control method is proposed, that determines a limited number of relevant worst-case scenarios on the fly, which makes the robust control optimization more suitable for real-time applications. The robust predictive traffic signal control method is implemented in the real-time prototyping environment as described in Chapter 5. The robust predictive traffic signal control method is analyzed for the most critical traffic situation and the most sensitive prediction model parameters identified in Chapter 3, i.e., a saturated corridor with uncertain turn fractions during peak periods that may suddenly increase the saturation degree of the corridor risking spillback and resulting in large performance loss in such worst-case scenarios. The results of the robust analysis show that:

2. *The heuristic robust predictive control approach is suitable for a real-time adaptive context and significantly reduces large performance loss due to prediction uncertainties in structure-free model-based predictive traffic signal control.* [Chapter 4]

Non-robust predictive control, based on the average turn fractions, can result in a large performance loss in the saturated corridor in worst-case scenarios (see Figure 4.3a). A prediction error due to the wrong assumption on the turn fractions can result in spillback blocking traffic at upstream intersections, as was already noted in the sensitivity analysis in 1B. The real-time predictive controller is too late to correct its decision and the damage is already done, even with a sufficient prediction horizon length, the structure-free control freedom, and highly adaptive frequent feedback mechanism. Applying robust control reduces the performance loss considerably (up to 60%) in the saturated corridor in worst-case scenarios (see Figure 4.3c). The robust controller identifies and anticipates on the worst-case scenarios in time, evaluating the full set of parameter uncertainty scenarios during the control optimization.

However, optimizing against the full scenario set is too time-consuming to perform in real-time. The heuristic robust approach proposed in this thesis reduces the scenario set to 12% of its full size in the robust optimization, saving calculation time while maintaining a similar control performance (see Figure 4.5). The heuristic approach makes robust control more suitable for real-time applications. Furthermore, the proposed robust control method turns out to be only temporarily overprotective, decreasing the average system performance only for a brief period (see Figure 4.4d). The robust control method is only applied in the periods that it is really needed, in this case during peak periods with uncertain turn fractions. Applying robust control shortly, only in times when needed, a safety margin is applied that leaves the traffic state in a less critical condition in the corridor such that, afterwards, the non-robust controller can control the traffic more adequately with less risk on spillback. Overall, the average system performance over all scenarios is increased in the saturated corridor (see Figure 4.3d). This answers research question 2 and shows the potential of real-time robust predictive traffic signal control, reducing the impact of prediction uncertainties on the control performance significantly.

6.2 Scientific contributions

The major contribution to the field of predictive traffic signal control is twofold, consisting of an analysis on prediction quality and a robust control design. This thesis provides new insights into the relation between prediction quality (accuracy) and control performance in a predictive traffic signal control system, which can be used in the design of predictive traffic signal control applications in real life. And this thesis provides a robust control approach, explicitly considering prediction uncertainties in the predictive traffic signal control system, proposing a new heuristic method to be applicable in a real-time adaptive context. This robust approach leads to improved predictive traffic signal control systems in which the impact of prediction uncertainties on the control performance is reduced. This thesis makes predictive signal control systems better applicable in real life, increasing control performance under prediction uncertainties.

6.2.1 Prediction quality

In Chapter 2 and 3 of this thesis a sensitivity analysis framework is proposed and new insights for control design are obtained regarding the influence of prediction errors on the behavior and performance of highly adaptive predictive traffic signal control systems:

- Theoretical framework for sensitivity analysis. [Chapter 2, 3]
  A sensitivity analysis framework is proposed, which aids researchers and practitioners to structurally study the effect of prediction errors on the control performance of predictive traffic signal control systems. Insights into performance loss due to current prediction errors and performance gain of possible prediction improvements indicate where to focus on in the control design.
- Insights into the effect of prediction errors. [Chapter 2, 3]
  The sensitivity analysis provides new insights into the relation between prediction quality and control performance for highly adaptive (structure-free) model-based predictive control (see key findings in Section 6.1.1). This thesis shows that, although prediction errors decrease control performance significantly, erroneous predictions are still useful in a highly adaptive predictive control system to efficiently control the traffic in a network.

- Design guidelines for predictive control under uncertainties. [Chapter 2, 3]
  Guidelines are provided to handle prediction errors properly, limiting the impact of prediction errors on the performance of the traffic signal control system. Summarized: A sufficient prediction horizon length is essential to connect information on multiple intersections, although covering all intersections is not necessary, since there is an optimal prediction horizon length after which performance does not improve anymore. A frequent reinitialization to the current state together with a high control freedom (preferably structure-free control) are essential to identify and correct prediction errors quickly. The current state is most important to estimate accurately to dampen the effect of other prediction errors in the control system. Saturation rates and turn fractions, although difficult to obtain in practice, are the most important model quantities to predict accurately, preferably including information on individual vehicles. These guidelines can be used in the design of real-life predictive traffic signal control systems managing prediction uncertainties.

6.2.2 Robust control

In Chapter 4 of this thesis a robust control approach is developed that explicitly considers prediction uncertainties in highly adaptive predictive traffic signal control systems to reduce the effect of prediction errors on the control performance:

- Robust control approach in highly adaptive context. [Chapter 4]
  A robust control approach is proposed for predictive traffic signal control systems to explicitly consider prediction uncertainties in the control design. Robust worst-case (minmax) scenario-based optimization techniques, originally developed for offline fixed-time control to protect against performance loss due to demand fluctuations, are translated to the new online highly adaptive context of (structure-free) model-based predictive traffic signal control to protect against performance loss due to uncertainties in the prediction model parameters. The robust control approach is made suitable for real-time applications (see heuristic control methods in Section 6.2.3).
- Insights into the potential of robust control approaches. [Chapter 4]
  This thesis provides new insights into the ability (and drawbacks) of robust control techniques for predictive traffic signal control systems to protect against performance loss due to prediction uncertainties (see key findings in Section 6.1.2). The proposed heuristic robust control method significantly reduces the impact of prediction errors on the control performance due to parameter uncertainties in highly adaptive (structure-free) model-based predictive traffic signal control. This thesis shows that robust control approaches are promising for predictive traffic signal control systems to improve the control performance under uncertainties.

6.2.3 Heuristic control methods

In Chapter 3 and 4 of this thesis new heuristic control methods are developed to make the robust and non-robust highly adaptive (structure-free) model-based predictive traffic signal controller suitable for real-time applications:

- Heuristic branch-and-bound control optimization. [Chapter 3]
  The highly adaptive (structure-free) model-based predictive controller considered in this thesis is a time-consuming discrete optimization problem. Problem complexity increases rapidly for networks consisting of multiple intersections. For the corridor studied in this thesis, control decisions are already difficult to optimize in real-time, especially in the short time (few seconds) available in the highly adaptive system with high update frequency. Therefore, a heuristic control optimization approach is proposed that comes up fast with close-to-optimal control decisions. A heuristic branch-and-bound control optimization is developed that approximates the tail of the decision sequence by greedy decisions, which are optimized further in the next decision update. The proposed heuristic has a much faster convergence than the exact method, reaching a suboptimal solution that lies within 1% control performance of the optimal solution (see Figure 3.3). The heuristic control optimization makes predictive traffic signal control more suitable for a network.
- Heuristic robust control approach. [Chapter 4]
  Robust control increases problem complexity further, including uncertainty scenarios in the highly adaptive predictive controller. Therefore, a heuristic robust control approach is developed that only uses a limited number of uncertainty scenarios, selecting the relevant scenarios on the fly during the control process. The heuristic robust approach reduces the scenario set to 12% of its full size in the robust optimization, saving calculation time while maintaining a similar control performance (see Figure 4.5). The heuristic robust control approach, in combination with the heuristic branch-and-bound control optimization, makes robust predictive traffic signal control more suitable for a real-time highly adaptive context.

6.2.4 Real-time prototyping

In Chapter 5, an open-source real-time prototyping environment is presented to design new types of highly adaptive predictive traffic signal control applications:

- Real-time prototyping environment. [Chapter 5]
  An environment for real-time prototyping is developed throughout the research in this thesis and is made available for future research. The real-time prototyping environment aids researchers and practitioners to design and analyze new types of advanced predictive traffic signal control methods. Predictive traffic signal control systems are studied as a similar process close to real-life but still by structural experiments in a controlled simulation environment. Requirements concerning real-time applicability are already considered in the control design, making the design step to real-life applications smaller.
- Insights into the potential of new control methods. [Chapter 2, 3]
  The real-time prototyping environment provides valuable insights into the potential of new types of advanced real-time highly adaptive predictive traffic signal control systems, as demonstrated throughout this thesis. Chapter 2 shows that highly adaptive structure-free predictive control outperforms the more traditional cyclic predictive control. Chapter 3 shows the performance benefit of global network control over local intersection control and the added value of including individual vehicle information in predictive traffic signal control systems. These new types of advanced real-time highly adaptive predictive traffic signal controllers have the potential to improve performance significantly in future traffic signal control applications.

6.3 Practical implications

In the design of real-life predictive traffic signal control systems, a real-time prototyping environment as developed in this thesis is essential to explore and test new control methods. Control methods can be developed and studied in a similar process close to real-life in an isolated and controlled simulation environment before the control systems are actually applied in real-life. In a real-time prototyping environment, not only theoretical algorithmic aspects are addressed, but also practical aspects like real-time feasibility, data exchange, and process synchronization (see Chapter 5). Practical drawbacks can be identified and solved in the early design phase of the control applications, designing suitable control systems for practice. The insights obtained in a real-time prototyping environment are most valuable for the design of traffic signal control applications in real-life, especially for the design of new types of control systems like the highly adaptive predictive controllers studied in this thesis.

For the real-life applicability of predictive traffic signal control systems, prediction uncertainties that occur in real-life processes need to be explicitly considered in the control design, as investigated in the prototyping environment as main aspect in this thesis. In addition, there are also other methodological aspects that must be included in the control design in order to apply predictive control systems in practice. The following methodological design aspects were noted during this research:

- Prediction uncertainties: In this thesis prediction uncertainties, as occur in a real-life process, are explicitly considered in the design of predictive traffic signal control methods. In the prototyping environment the effect on the control performance of prediction uncertainties is analyzed and robust techniques are investigated to reduce the impact of the uncertainties (see Section 6.1). Based on the analysis in the prototyping environment, guidelines are provided for the design of predictive traffic signal control systems under uncertainties in real-life (see Section 6.2.1) and a heuristic robust control method is proposed to explicitly include prediction uncertainties in the real-life control design (see Section 6.2.2). The impact of uncertainties on the control performance and methods to reduce this impact can be explored further in the prototyping environment, for different network configurations, traffic demand patterns, and uncertainty scenarios, focused on the traffic signal control application that is under design. Managing uncertainties properly in the traffic signal control design, as in this thesis, reduces performance loss due to uncertainties and increases the real-life applicability of the control system.

- Real-time applicability: In real-life traffic signal control applications, the control decisions need to be calculated in real-time. Problem complexity and calculation times need to be considered in the control design. The (robust) structure-free model-based predictive traffic signal controller in this thesis already is a complex discrete optimization problem with a time-consuming solution process. Therefore, heuristic solution approaches are proposed (see Section 6.2.3), that come up fast with close-to-optimal solutions. In real-life control applications, the problem complexity will increase further, considering larger networks with more intersections, increasing the prediction horizon, including more movement groups also for bikes and pedestrians, switching signals more often (e.g., each second), using more detailed vehicle information, and considering more uncertainty scenarios in case of robust control. Additional methods need to be developed to reduce calculation times of predictive control methods, like, next to heuristic approaches aiming at near optimality, parametrization of control variables, restriction of the search space adding constraints, hierarchical approaches splitting up the network, parallelization techniques and hardware-based solutions. In the prototyping environment the speed and quality of the designed control method can be analyzed under real-time restrictions. Reducing calculation times in the design of the control method increases the real-life applicability of the control system, especially towards large-scale network-wide predictive control applications.

- Control constraints: In this thesis, the focus is on structure-free control, providing a large control freedom without imposing cycles and additional constraints. In practice, however, there are many constraints that need to be considered in the control design. Some constraints are a necessity, like the priority of emergency vehicles. Some constraints are imposed by safety, like preventing collisions and passages through red by sufficient clearance times, yellow times, and green times, and preventing the blockages of off-ramps, bridges, and railway crossings. Other constraints may arise from policies, like the priority of public transport vehicles, bikes or pedestrians, preferences on route guidance, and the acceptable maximum waiting (or red) times. And some constraints may prevail from common practice, like the traditional (and proven) cyclic control structure. This thesis shows that cyclic constraints in predictive traffic signal control significantly limit the control performance (see Section 6.1.1). If too many constraints are added to the predictive controller, then the optimization may not be beneficial anymore, lacking degrees of control freedom. In the design process of the predictive controller, the necessity and desirability of the different, and sometimes conflicting, constraints need to be considered. The prototyping environment can provide the necessary insights into the effects of additional constraints on the control performance. Weighing the pros and cons of the different constraints in the control design, the designer can clearly define and substantiate the boundaries of the control system for real-life applications.

- Control objectives: In this thesis, the predictive control system aims to reduce congestion by minimizing the delay of queued vehicles. In practice, different traffic management goals may be pursued arising from different policies, considering not only throughput, but also other aspects like safety and environmental issues or the comfort for road users. Besides, the traffic management goal on a network level may differ from the goals on the local intersection level. In the design of the predictive control system, all these (partly conflicting) objectives need to be combined, resulting in a multi-criteria optimization. In a multi-criteria optimization, control decisions with an almost equal close-to-optimal performance on one criterium, e.g., throughput, can be reviewed on another criterium, like the stability of the control decisions providing more comfort for road users. The prototyping environment can provide the necessary insights into the trade-off between the different performance objectives. Combining the different objectives properly in the control design, the designer can customize the control system to fulfill the goal of real-life applications.

These (and probably other) methodological aspects need to be considered in the design of predictive traffic signal control and can be analyzed further using the real-time prototyping environment in this thesis as a starting point. Next to the development and testing of the predictive control method, the acceptance of the new type of control system needs to be considered. Note that the highly adaptive structure-free predictive control method in this thesis deviates from the traditional cyclic control approach. The structure-free controller gives green to the different movements in arbitrary order and combination, letting go the traditional cycles no longer giving green to the different movements in the same repeating order. The introduction of structure-free control may lead to misunderstanding among road users due to non-matching expectations and may be experienced as unsafe or may actually cause unsafe situations. The introduction of predictive traffic signal control in general may also ask for a mental shift among policy makers, letting go some of the control constraints making full use of the new possibilities of the more flexible multi-objective predictive control system. For a successful widely supported implementation, the acceptance by policy makers and road users is a must, together with the technological production aspects (which will not be considered in this thesis). This thesis improves the real-life applicability of highly adaptive predictive control methods by reducing the impact of uncertainties, considering one of the main practical issues. However, for a successful application of these control systems in real-life, the whole process from control design (in a wider perspective), prototyping (testing), acceptance and production need to be covered.

6.4 Scientific recommendations

Considering the research in this thesis on uncertainties in predictive traffic signal control systems, the following scientific recommendations can be made for the further development of predictive traffic signal control methods:

- Multi-dimensional sensitivity analysis: In the one-dimensional sensitivity analysis in this thesis, quantities in the predictive traffic signal control system are varied one-by-one. Although the prediction errors in the different quantities cause a significant performance loss, prediction is still beneficial, i.e., an increasing prediction horizon still increases control performance even if the predicted quantities contain errors. The predictive property of the traffic signal control system is strong and remains preserved under errors, since other undisturbed quantities are available, and enough structure is left in the prediction model to rely on. Future sensitivity analysis may focus on a combination of errors in the different quantities at the same time, which may amplify each other causing a larger performance loss in a network. The multi-dimensional sensitivity analysis may identify the tipping point (if there is any) when a prediction model has insufficient structure left and prediction is not beneficial anymore.

- Hybrid prediction methods: The sensitivity analysis in this thesis shows that turn fractions (vehicle directions) and saturation rates (vehicle departures) are the most sensitive quantities in the predictive traffic signal control system in a network. Improving the prediction accuracy for these quantities, preferably by using individual vehicle information, increases control performance considerably. Therefore, future research should focus on improving traffic state estimation and prediction methods by including individual vehicle information, and on improving the underlying data quality of real-time vehicle information. Prediction methods may become hybrid, containing micro vehicle-based (and data-driven) predictions for the available connected vehicle information, in combination with macro flow-based (and model-based) predictions for a complete traffic state.

- Robust adaptive control: In this thesis robust control is applied in the real-time adaptive context of predictive traffic signal control systems. Identifying worst-cases when the traffic situation becomes too critical, and maintaining the traffic state in less critical conditions, is essential for robust control. In future research, this can possibly be improved in real-time adaptive robust control by considering end costs in the predictive controller, which represents the remaining costs that fall outside the scope of the prediction horizon. Besides, a more representative control objective can potentially be used that indicates explicitly the risk on a too critical traffic condition, e.g., densities that indicate the risk on spillback. Moreover, robust control may be switched on and off automatically based on traffic conditions, applying robust control only in near critical conditions. In this way, an adaptive robust controller can be developed that identifies worst-case scenarios better and sooner, protecting only when it is needed (limiting overprotectiveness).

- Hierarchical network control: In this thesis predictive traffic signal control is studied on a corridor. For city-wide traffic management, predictive control methods need to be scaled up to a network-wide control application. Future research should focus on a network-wide control optimization, providing solutions for the increasing problem complexity of predictive control in larger networks. A hierarchical approach may be followed, splitting up the network and decoupling intersections. On the upper level, the overall network traffic management goals should be pursued, on the lower level, clusters of intersections may be controlled locally with their own goals contributing to the common good. Research is needed on a ‘robust’ decoupling of intersections. The network controller should use robust approaches to prevent critical traffic conditions in the overall network, such that intersections can be decoupled safely and have enough control freedom to be controlled locally. A robust hierarchical control approach (together with heuristic optimization techniques) could make predictive traffic signal control suitable for large-scale network-wide applications.

These developments can lead to network-wide robust predictive traffic signal control applications, an essential part of the traffic management system to reduce congestion in urban areas. These future predictive control applications can be highly adaptive making full use of new available detailed predictions of individual vehicle patterns, increasing control performance in a network. This thesis contributes to this future perspective, taking a step towards highly adaptive predictive control systems that suffer less from prediction uncertainties and can fully benefit from traffic predictions.

Bibliography

[1] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, “Review of road traffic control strategies,” *Proc. IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.

[2] M. Eom and B.-I. Kim, “The traffic signal control problem for intersections: a review,” *Eur. Transp. Res. Rev.*, vol. 12, 50, 2020.

[3] Y. Wang, X. Yang, H. Liang, and Y. Liu, “A review of the self-adaptive traffic signal control system based on future traffic environment,” *J. Adv. Transp.*, vol. 2018, 1096123, 2018.

[4] Q. Guo, L. Li, and X. Ban, “Urban traffic signal control with connected and automated vehicles: A survey,” *Transp. Res. Part C Emerg. Technol.*, vol. 101, pp. 313–334, 2019.

[5] L. Li, D. Wen, and D. Yao, “A Survey of Traffic Control With Vehicular Communications,” *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 425–432, 2014.

[6] X. Wu and H. X. Liu, “Using high-resolution event-based data for traffic modeling and control: An overview,” *Transp. Res. Part C Emerg. Technol.*, vol. 42, pp. 28–43, 2014.

[7] V. Astarita, V. P. Giofrè, G. Guido, and A. Vitale, “The use of adaptive traffic signal systems based on floating car data,” *Wirel. Commun. Mob. Comput.*, 2017.

[8] T. Seo, A. M. Bayen, T. Kusakabe, and Y. Asakura, “Traffic state estimation on highway: A comprehensive survey,” *Annu. Rev. Control*, vol. 43, pp. 128–151, 2017.

[9] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, “Short-term traffic forecasting: Where we are and where we’re going,” *Transp. Res. Part C Emerg. Technol.*, vol. 43, pp. 3–19, 2014.

[10] J. W. C. Van Lint and C. P. I. J. Van Hinsbergen, “Short-term traffic and travel time prediction models,” *Artif. Intell. Appl. to Crit. Transp. Issues*, vol. 22, no. 1, pp. 22–41, 2012.

[11] R. T. Van Katwijk, “Multi-Agent Look-Ahead Traffic-Adaptive Control,” TRAIL Thesis Series T2008/3, The Netherlands TRAIL Research School, Delft, 2008.

[12] A. Wilson, *Handboek verkeerslichtenregelingen 2014*. CROW, 2014.

[13] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, "Short-term traffic forecasting: Overview of objectives and methods," *Transp. Rev.*, vol. 24, no. 5, pp. 533–557, 2004.

[14] I. Lana, J. Del Ser, M. Velez, and E. I. Vlahogianni, "Road Traffic Forecasting: Recent Advances and New Challenges," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 2, pp. 93–109, 2018.

[15] M. G. H. Bell and Y. Iida, *Transportation network analysis*. Wiley, 1997.

[16] E. Cascetta, *Transportation systems analysis: models and applications*. Springer optimization and its applications, vol. 29, Springer, 2009.

[17] J. de Dios Ortúzar and L. G. Willumsen, *Modelling transport*. Wiley, 4th Edition, 2011.

[18] A. Stathopoulos and M. G. Karlaftis, "A multivariate state space approach for urban traffic flow modeling and prediction," *Transp. Res. Part C Emerg. Technol.*, vol. 11, no. 2, pp. 121–135, 2003.

[19] M. Burger, M. van den Berg, A. Hegyi, B. De Schutter, and J. Hellendoorn, "Considerations for model-based traffic control," *Transp. Res. Part C Emerg. Technol.*, vol. 35, pp. 1–19, 2013.

[20] K. Aboudolas, M. Papageorgiou, and E. Kosmatopoulos, "Store-and-forward based methods for the signal control problem in large-scale congested urban road networks," *Transp. Res. Part C Emerg. Technol.*, vol. 17, no. 2, pp. 163–174, 2009.

[21] 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," *Transp. Res. Part C Emerg. Technol.*, vol. 18, no. 5, pp. 680–694, 2010.

[22] G. S. de Weg, M. Keyvan-Ekbatani, A. Hegyi, and S. P. Hoogendoorn, "Urban network throughput optimization via model predictive control using the link transmission model," *Proc. 95th Annu. Meet. Transp. Res. board*, pp. 1–19, 2016.

[23] C. Ledoux, "An urban traffic flow model integrating neural networks," *Transp. Res. Part C Emerg. Technol.*, vol. 5, no. 5, pp. 287–300, 1997.

[24] B.-L. Ye *et al.*, "A survey of model predictive control methods for traffic signal control," *IEEE/CAA J. Autom. Sin.*, vol. 6, no. 3, pp. 623–640, 2019.

[25] C. E. García, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[26] E. F. Camacho and C. Bordons, *Model predictive control in the process industry*. Springer, 1995.

[27] J. M. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, 2002.

[28] G. A. Klunder, H. Taale, L. Kester, and S. Hoogendoorn, "The Effect of Inaccurate Traffic Data for Ramp Metering: Comparing Loop Detectors and Cameras Using Information Utility," *IFAC Proc. Vol.*, vol. 47, no. 3, pp. 11318–11325, 2014.

[29] T. Tettamanti, I. Varga, T. Péni, T. Luspay, and B. Kulcsár, "Uncertainty modeling and robust control in urban traffic," *IFAC Proc. Vol.*, vol. 44, no. 1, pp. 14910–14915, 2011.

[30] B.-L. Ye, W. Wu, H. Gao, Y. Lu, Q. Cao, and L. Zhu, "Stochastic Model Predictive Control for Urban Traffic Networks," *Appl. Sci.*, vol. 7, no. 6, 2017.

[31] Y. Yin, "Robust optimal traffic signal timing," *Transp. Res. Part B Methodol.*, vol. 42, no. 10, pp. 911–924, 2008.

[32] J.-Q. Li, "Discretization modeling, integer programming formulations and dynamic programming algorithms for robust traffic signal timing," *Transp. Res. Part C Emerg. Technol.*, vol. 19, no. 4, pp. 708–719, 2011.

[33] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization*. Princeton series in applied mathematics, vol. 28, Princeton university press, 2009.

[34] T. Tettamanti, T. Luspay, B. Kulcsár, T. Péni, and I. Varga, "Robust Control for Urban Road Traffic Networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 385–398, 2014.

[35] B. Waterson and S. Box, "Quantifying the impact of probe vehicle localisation data errors on signalised junction control," *IET Intell. Transp. Syst.*, vol. 6, no. 2, pp. 197–203, 2012.

[36] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn, "Efficient network-wide model-based predictive control for urban traffic networks," *Transp. Res. Part C Emerg. Technol.*, vol. 24, pp. 122–140, 2012.

[37] Aimsun, "Aimsun 8.2 User's Manual." Aimsun Version 8.2.0, Barcelona, Spain, 2017.

[38] L. Zhang, Y. Yin, and Y. Lou, "Robust Signal Timing for Arterials under Day-to-Day Demand Variations," *Transp. Res. Rec.*, vol. 2192, no. 1, pp. 156–166, 2010.

[39] S. V Ukkusuri, G. Ramadurai, and G. Patil, "A robust transportation signal control problem accounting for traffic dynamics," *Comput. Oper. Res.*, vol. 37, no. 5, pp. 869–879, 2010.

[40] B. Park, I. Yun, and K. Ahn, "Stochastic Optimization for Sustainable Traffic Signal Control," *Int. J. Sustain. Transp.*, vol. 3, no. 4, pp. 263–284, 2009.

[41] J. Stevanovic, A. Stevanovic, P. T. Martin, and T. Bauer, "Stochastic optimization of traffic control and transit priority settings in VISSIM," *Transp. Res. Part C Emerg. Technol.*, vol. 16, no. 3, pp. 332–349, 2008.

[42] C. Osorio and L. Chong, "A Computationally Efficient Simulation-Based Optimization Algorithm for Large-Scale Urban Transportation Problems," *Transp. Sci.*, vol. 49, no. 3, pp. 623–636, 2015.

[43] L. K. Jones, R. Deshpande, N. H. Gartner, C. Stamatiadis, and F. Zou, "Robust controls for traffic networks: The near-Bayes near-Minimax strategy," *Transp. Res. Part C Emerg. Technol.*, vol. 27, pp. 205–218, 2013.

[44] B. Park and A. Kamarajugadda, “Development and Evaluation of a Stochastic Traffic Signal Optimization Method,” *Int. J. Sustain. Transp.*, vol. 1, no. 3, pp. 193–207, 2007.

[45] K. Wada, K. Usui, T. Takigawa, and M. Kuwahara, “An optimization modeling of coordinated traffic signal control based on the variational theory and its stochastic extension,” *Transp. Res. Part B Methodol.*, vol. 117, pp. 907–925, 2018.

[46] H. Liu, K. Han, V. V Gayah, T. L. Friesz, and T. Yao, “Data-driven linear decision rule approach for distributionally robust optimization of on-line signal control,” *Transp. Res. Part C Emerg. Technol.*, vol. 59, pp. 260–277, 2015.

[47] F. V. Webster, “Traffic signal settings,” Road Research Technical Paper No 39, Department of Scientific and Industrial Research, Road Research Laboratory, London, UK, 1958.

[48] W.-H. Lin, H. K. Lo, and L. Xiao, “A Quasi-Dynamic Robust Control Scheme for Signalized Intersections,” *J. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 223–233, 2011.

[49] S. Liu *et al.*, “Robust receding horizon parameterized control for multi-class freeway networks: A tractable scenario-based approach,” *Int. J. Robust Nonlinear Control*, vol. 26, no. 6, pp. 1211–1245, 2016.

[50] X.-H. Yu and W. W. Recker, “Stochastic adaptive control model for traffic signal systems,” *Transp. Res. Part C Emerg. Technol.*, vol. 14, no. 4, pp. 263–282, 2006.

[51] Y. Tong, L. Zhao, L. Li, and Y. Zhang, “Stochastic programming model for oversaturated intersection signal timing,” *Transp. Res. Part C Emerg. Technol.*, vol. 58, pp. 474–486, 2015.

[52] W. Ma, K. An, and H. K. Lo, “Multi-stage stochastic program to optimize signal timings under coordinated adaptive control,” *Transp. Res. Part C Emerg. Technol.*, vol. 72, pp. 342–359, 2016.

Summary

Traffic signal control is an important part of the traffic management system to reduce congestion in urban areas. Traffic signal control systems are becoming more advanced. New technologies in the field of communicating vehicles and intelligent traffic lights make the traffic signal control system more interactive. New real-time data sources on individual vehicles become available to provide the traffic signal control system with improved insights into actual traffic conditions. Advanced prediction methods are developed that even make it possible to pro-actively control traffic based on predicted traffic conditions. In general, traffic signal controllers are becoming more adaptive, reconsidering control decisions more often, using more information, and allowing more degrees of control freedom, such that the control decision matches the upcoming traffic pattern better up to the level of individual vehicle movements. These advanced highly adaptive predictive traffic signal control systems may have the potential to efficiently control traffic in a network increasing control performance.

However, since these control systems rely on more information, highly adaptive predictive traffic signal controllers may be vulnerable to information errors that occur in practice. In real life, prediction methods are not perfect and will contain errors. These prediction errors will eventually affect the control decision and the performance of the control system. To fully benefit from prediction techniques in traffic signal control systems in practice, the performance impact of prediction uncertainties needs to be addressed in the control design. Until now, prediction uncertainties have been barely considered in existing predictive traffic signal control methods. Therefore, in this thesis, highly adaptive predictive traffic signal control systems are studied considering prediction uncertainties. In specific, the effect of prediction errors on the control performance is analyzed and robust methods to reduce the performance impact of prediction uncertainties are designed.

A prototyping environment is developed, and is made available, to study real-time highly adaptive predictive traffic signal controllers under uncertainties. An existing microscopic simulator is used as a basis, and an adaptive (structure-free) model-based predictive traffic signal controller is designed as a plugin, both running in parallel and communicating in real time. The predictive control design in the real-time prototyping environment already addresses real-life requirements on information exchange, process synchronization and real-time applicability, making the design step to real-life control applications smaller. A sensitivity analysis is performed in the prototyping environment for the adaptive (structure-free) model-based predictive traffic signal control system to study the effect of prediction errors on the control performance. To this end, first the ideal world with perfect predictions is created in the

microscopic simulator, in which the predictive traffic signal controller can reach its potential performance. Then, prediction errors are structurally introduced one-by-one in the different model quantities of the control system, and the performance loss due to these errors is analyzed. The sensitivity analysis is performed for local control on a single intersection and is extended to network control on a corridor of multiple intersections. Furthermore, a heuristic robust control approach is designed in the prototyping environment for the real-time adaptive predictive control context to protect against the performance loss due to prediction uncertainties. The robust method is analyzed on the ability to reduce performance loss in critical traffic situations in the corridor for uncertainties in the most sensitive model parameters in the predictive control system.

This thesis shows that prediction errors can decrease the control performance significantly for highly adaptive (structure-free) model-based predictive traffic signal control systems. However, the erroneous predictions are still beneficial for the control system, i.e., an increase of the prediction horizon improves the control performance although the prediction contains errors. For the highly adaptive predictive traffic signal controller studied under perfect predictions, a longer prediction horizon leads to an increase in control performance up to a certain optimal prediction horizon length. This predictive property of the control system remains preserved under erroneous predictions, since enough remaining structure is left in the prediction model to rely on. Especially in a saturated network, connecting intersections by predictive information, even when it contains errors, is essential to control traffic efficiently. The high adaptivity of the control system, i.e., a high update frequency to reinitialize often to the current state by the feedback mechanism and a large control freedom (preferably structure-free control) to correct mistakes quickly, is important to limit the impact of prediction errors. However, in critical traffic conditions, a significant performance loss remains due to prediction errors caused by input errors, aggregation errors, and biases in the prediction model, especially for the most sensitive quantities, i.e., saturation rates and turn fractions. So, more accurate predictions for these most sensitive quantities, and the use of disaggregated individual vehicle data, would improve control performance in future control applications. However, these quantities may also be the most difficult to predict in practice, containing the largest uncertainties. Uncertainties in these prediction model parameters should therefore be considered explicitly in a more robust control design. This thesis shows that robust control approaches can be made suitable for real-time adaptive (structure-free) model-based predictive control and reduce the performance loss due to prediction uncertainties considerably.

This thesis improves the real-life applicability of highly adaptive predictive traffic signal control methods by reducing the impact of uncertainties, addressing one of the main practical issues. Other practical design aspects need to be considered as well, like the real-time applicability for large-scale network-wide control applications, the impact of control constraints on the control freedom of the highly adaptive system, and the trade-off in multi-criteria control objectives for different traffic management policies. Moreover, for a successful application of these new types of control systems in real-life, the whole process from control design, prototyping, acceptance, and production needs to be covered. This thesis contributes to the future perspective of pro-active network-wide traffic management and control, taking a step towards more robust highly adaptive predictive traffic signal control systems that suffer less from prediction uncertainties and can fully benefit from traffic predictions.

Samenvatting

Verkeerslichtenregelingen zijn een belangrijk onderdeel van het verkeersmanagementsysteem om congestie in stedelijke gebieden te reduceren. Verkeersregelsystemen worden steeds geavanceerder. Nieuwe technologieën op het gebied van communicerende voertuigen en intelligente verkeerslichten maken het verkeersregelsysteem interactiever. Nieuwe realtime databronnen van individuele voertuigen komen beschikbaar die meer inzicht geven in de actuele verkeerssituatie. Geavanceerde voorspelmethoden worden ontwikkeld die het zelfs mogelijk maken om het verkeer proactief te regelen op basis van de voorspelde verkeerssituatie. In het algemeen worden verkeerslichtenregelingen steeds adaptiever, waarbij beslissingen vaker worden heroverwogen, meer informatie wordt gebruikt, en meer regelvrijheid wordt toegestaan, zodat de regeling beter aansluit bij het actuele verkeerspatroon en individuele voertuigbewegingen. Deze geavanceerde, hoogadaptieve, voorspellende verkeerslichtenregelingen hebben potentie om verkeer in een netwerk efficiënt te regelen en verkeersregelprestaties te verhogen.

Echter, deze hoogadaptieve, voorspellende verkeerslichtenregelingen zijn sterk afhankelijk van informatie, en kunnen kwetsbaar zijn voor informatiefouten die in de praktijk optreden. Voorspelmethoden zijn niet perfect en voorspellingen kunnen fouten bevatten. Deze voorspelfouten zullen uiteindelijk van invloed zijn op de beslissing en de prestaties van het verkeersregelsysteem. Om optimaal te kunnen profiteren van voorspeltechnieken in verkeersregelsystemen, moet de invloed van voorspelonzekerheden op de regelprestaties worden meegenomen in het ontwerp. Tot nu toe houden voorspellende verkeersregelmethoden nog nauwelijks rekening met onzekerheden. Daarom worden in dit proefschrift hoogadaptieve voorspellende verkeerslichtenregelingen bestudeerd, waarbij voorspelonzekerheden worden meegenomen. In het bijzonder worden de effecten van voorspelfouten op de regelprestatie geanalyseerd en worden robuuste regelsystemen ontworpen om de impact van voorspelonzekerheden te reduceren.

Speciaal voor dit proefschrift is een opensource prototypingomgeving ontwikkeld om realtime hoogadaptieve voorspellende verkeerslichtenregelingen onder onzekerheden te bestuderen. Een bestaande microscopische simulator is gebruikt als basis en een adaptieve (structuurvrije) modelgebaseerde voorspellende verkeerslichtenregeling is ontworpen als parallelle plug-in. Het regelontwerp in de prototypingomgeving voldoet al aan praktijkeisen op het gebied van informatieuitwisseling, processynchronisatie en realtime toepasbaarheid, waardoor de ontwerpstap naar de praktijk kleiner wordt. In de prototypingomgeving wordt een gevoeligheidsanalyse uitgevoerd voor het hoogadaptieve (structuurvrije) modelgebaseerde voorspellende verkeersregelsysteem om het effect van voorspelfouten op de regelprestatie te bestuderen. Daartoe wordt in de microscopische simulator eerst de ideale wereld met perfecte voorspellingen gecreëerd, waarin de voorspellende verkeerslichtenregeling zijn potentiële prestatie kan bereiken. Vervol-

gens worden één voor één voorspelfouten geïntroduceerd in de verschillende modelgrootheden van het verkeersregelsysteem en wordt het prestatieverlies als gevolg van deze fouten geanalyseerd. De gevoeligheidsanalyse wordt uitgevoerd voor een lokale regeling op een enkel kruispunt en uitgebreid naar een netwerkregeling op een corridor van meerdere kruispunten. Bovendien wordt er in de prototypingomgeving een heuristische robuuste methode ontworpen voor het realtime adaptieve voorspellende regelsysteem om te beschermen tegen prestatieverlies als gevolg van voorspelonzekerheden. De robuuste methode wordt geanalyseerd op het vermogen om prestatieverlies in kritieke verkeerssituaties in de corridor te reduceren voor onzekerheden in de meest gevoelige modelparameters in het voorspellende verkeersregelsysteem.

Dit proefschrift laat zien dat door voorspelfouten de regelprestaties aanzienlijk kunnen afnemen voor hoogadaptieve (structuurvrije) modelgebaseerde voorspellende verkeerslichtenregelingen. De foutieve voorspellingen zijn echter nog steeds nuttig voor het verkeersregelsysteem, d.w.z. het vergrootten van de voorspelhorizon verbetert de regelprestaties, zelfs als de voorspelling fouten bevat. Bij perfecte voorspellingen leidt een langere voorspelhorizon tot een toename van de regelprestatie tot een bepaalde optimale lengte van de voorspelhorizon. Deze voorspellende eigenschap van het regelsysteem blijft behouden bij foutieve voorspellingen, omdat er in het voorspelmodel voldoende resterende structuur overblijft. Vooral in een verzadigd netwerk is het verbinden van kruispunten door voorspellende informatie, zelfs als deze fouten bevat, essentieel om het verkeer efficiënt te regelen. De hoge adaptiviteit van het regelsysteem, d.w.z. een hoge updatefrequentie om met het feedbackmechanisme vaak te herinitialiseren naar de actuele verkeerstoestand en een grote regelvrijheid (bij voorkeur structuurvrije regeling) om fouten snel te corrigeren, is belangrijk om de impact van voorspelfouten te beperken. In kritieke verkeerssituaties blijft er echter een aanzienlijk prestatieverlies over door voorspelfouten als gevolg van invoerfouten, aggregatiefouten en biases in het voorspelmodel, met name voor de meest gevoelige grootheden, d.w.z. afrijdcapaciteiten en afslagpercentages. Nauwkeurigere voorspellingen voor deze meest gevoelige grootheden en het gebruik van gedesaggregeerde individuele voertuigdata zouden de regelprestaties van toekomstige verkeersregelsystemen kunnen verbeteren. Echter, deze grootheden zijn in de praktijk waarschijnlijk ook het moeilijkst te voorspellen, omdat ze de grootste onzekerheden bevatten. Onzekerheden in deze parameters van het voorspelmodel moeten daarom expliciet meegenomen worden in een robuuster regelontwerp. Dit proefschrift laat zien dat robuuste methoden ingezet kunnen worden voor realtime adaptieve (structuurvrije) modelgebaseerde voorspellende verkeerslichtenregelingen en het prestatieverlies als gevolg van voorspelonzekerheden aanzienlijk kunnen reduceren.

Dit proefschrift verbetert de praktische toepasbaarheid van hoogadaptieve voorspellende verkeerslichtenregelingen door de impact van onzekerheden te reduceren, waarmee een van de belangrijkste praktische problemen wordt aangepakt. In de praktijk moet echter ook rekening worden gehouden met andere ontwerpaspecten, zoals de realtime toepasbaarheid voor grootschalige netwerkregelingen, de invloed van beleidsbeperkingen op de regelvrijheid van het hoogadaptieve systeem, en de afweging tussen verschillende beleidsdoelstellingen. Voor een succesvolle toepassing van deze nieuwe geavanceerde regelsystemen moet bovendien het hele proces van regelontwerp, prototyping, acceptatie en productie worden doorlopen. Dit proefschrift draagt bij aan het toekomstperspectief van proactief netwerkbreed verkeersmanagement, waarbij een stap wordt gezet naar robuustere, hoogadaptieve voorspellende verkeerslichtenregelingen die minder last hebben van voorspelonzekerheden en ten volle kunnen profiteren van verkeersvoorspellingen.

About the author

Muriel Celeste Poelman was born on the 9th of January 1979 in Arnhem in the Netherlands. She attended secondary school at Het Gelders College in Arnhem, where she completed VWO-Gymnasium in 1997. After secondary school, Muriel studied Applied Mathematics at Delft University of Technology. She specialized in operations research, applied in traffic modeling, and received her MSc degree cum laude in 2004.

Muriel started her professional career in 2004 as a scientific researcher at TNO in Delft, working on the development and application of traffic models. She switched jobs in 2007 to become a traffic engineer at Royal HaskoningDHV in Amersfoort. There, she specialized further in traffic modeling applications on macro-, meso-, and microscopic scale, and worked together with Aimsun in the model development of their, equally named, traffic simulation software. Over time, she became more and more interested in real-time traffic modeling, and the use of short-term traffic predictions in traffic management and control applications. Following this interest, Muriel started her PhD research, part-time, at Delft University of Technology in April 2018. In this PhD research, she combined her background on operations research and her experience on traffic modeling, into the challenging field of predictive traffic signal control. After finishing her PhD research, Muriel will continue to work at Royal HaskoningDHV in the development of the traffic signal control software Flowtack, to apply her newly acquired knowledge on predictive control in practice.

Journal Articles

M. C. Poelman, A. Hegyi, A. Verbraeck, and J. W. C. van Lint, “Sensitivity Analysis to Define Guidelines for Predictive Control Design,” *Transp. Res. Rec.*, vol. 2674, no. 6, pp. 385–398, 2020. https://doi.org/10.1177/0361198120919114

M. C. Poelman, A. Hegyi, A. Verbraeck, and J. W. C. van Lint, “Structure-free model-based predictive signal control: A sensitivity analysis on a corridor with spillback,” *Transp. Res. Part C Emerg. Technol.*, vol. 153, 104174, 2023. https://doi.org/10.1016/j.trc.2023.104174

Peer-reviewed Conference Contributions

M. C. Poelman, A. Hegyi, A. Verbraeck, and J. W. C. van Lint, “Sensitivity Analysis on Information Quality for Signalized Traffic Control,” *8th Symp. Eur. Assoc. Res. Transp.*, Budapest, 2019.

M. C. Poelman, A. Hegyi, A. Verbraeck, and J. W. C. van Lint, “Sensitivity Analysis to Define Guidelines for Predictive Control Design,” *TRB 99th Annual Meeting*, Washington D.C., 2020.

Articles under Review

M. C. Poelman, A. Hegyi, A. Verbraeck, and J. W. C. van Lint, “Robust approach for structure-free model-based predictive traffic signal control considering uncertainties in model parameters,” *Under review*.

TRAIL Thesis Series

The following list contains the most recent dissertations in the TRAIL Thesis Series. For a complete overview of more than 400 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.

Poelman, M.C., *Predictive Traffic Signal Control under Uncertainty: Analyzing and Reducing the Impact of Prediction Errors,* T2024/6, October 2024, TRAIL Thesis Series, the Netherlands

Berge, S.H., *Cycling in the age of automation: Enhancing cyclist interaction with automated vehicles through human-machine interfaces,* T2024/5, September 2024, TRAIL Thesis Series, the Netherlands

Wu, K., *Decision-Making and Coordination in Green Supply Chains with Asymmetric Information,* T2024/4, July 2024, TRAIL Thesis Series, the Netherlands

Wijnen, W., *Road Safety and Welfare,* T2024/3, May 2024, TRAIL Thesis Series, the Netherlands

Caiati, V., *Understanding and Modelling Individual Preferences for Mobility as a Service,* T2024/2, March 2024, TRAIL Thesis Series, the Netherlands

Vos, J., *Drivers' Behaviour on Freeway Curve Approach,* T2024/1, February 2024, TRAIL Thesis Series, the Netherlands

Geržinič, N., *The Impact of Public Transport Disruptors on Travel Behaviour,* T2023/20, December 2023, TRAIL Thesis Series, the Netherlands

Dubey, S., *A Flexible Behavioral Framework to Model Mobility-on-Demand Service Choice Preference,* T2023/19, November 2023, TRAIL Thesis Series, the Netherlands

Sharma, S., *On-trip Behavior of Truck Drivers on Freeways: New mathematical models and control methods,* T2023/18, October 2023, TRAIL Thesis Series, the Netherlands

Ashkrof, P., *Supply-side Behavioural Dynamics and Operations of Ride-sourcing Platforms,* T2023/17, October 2023, TRAIL Thesis Series, the Netherlands

Sun, D., *Multi-level and Learning-based Model Predictive Control for Traffic Management,* T2023/16, October 2023, TRAIL Thesis Series, the Netherlands

Brederode, L.J.N., *Incorporating Congestion Phenomena into Large Scale Strategic Transport Model Systems*, T2023/15, October 2023, TRAIL Thesis Series, the Netherlands

Hernandez, J.I., *Data-driven Methods to study Individual Choice Behaviour: with applications to discrete choice experiments and Participatory Value Evaluation experiments*, T2023/14, October 2023, TRAIL Thesis Series, the Netherlands

Aoun, J., *Impact Assessment of Train-Centric Rail Signaling Technologies,* T2023/13, October 2023, TRAIL Thesis Series, the Netherlands

Pot, F.J., *The Extra Mile: Perceived accessibility in rural areas*, T2023/12, September 2023, TRAIL Thesis Series, the Netherlands

Nikghadam, S., *Cooperation between Vessel Service Providers for Port Call Performance Improvement,* T2023/11, July 2023, TRAIL Thesis Series, the Netherlands

Li, M., *Towards Closed-loop Maintenance Logistics for Offshore Wind Farms: Approaches for strategic and tactical decision-making*, T2023/10, July 2023, TRAIL Thesis Series, the Netherlands

Berg, T. van den, *Moral Values, Behaviour, and the Self: An empirical and conceptual analysis,* T2023/9, May 2023, TRAIL Thesis Series, the Netherlands

Shelat, S., *Route Choice Behaviour under Uncertainty in Public Transport Networks: Stated and revealed preference analyses,* T2023/8, June 2023, TRAIL Thesis Series, the Netherlands

Zhang, Y., Flexible, *Dynamic, and Collaborative Synchromodal Transport Planning Considering Preferences*, T2023/7, June 2023, TRAIL Thesis Series, the Netherlands

Kapetanović, M., *Improving Environmental Sustainability of Regional Railway Services,* T2023/6, June 2023, TRAIL Thesis Series, the Netherlands

Li, G., *Uncertainty Quantification and Predictability Analysis for Traffic Forecasting at Multiple Scales,* T2023/5, April 2023, TRAIL Thesis Series, the Netherlands

Harter, C., *Vulnerability through Vertical Collaboration in Transportation: A complex networks approach,* T2023/4, March 2023, TRAIL Thesis Series, the Netherlands

Razmi Rad, S., *Design and Evaluation of Dedicated Lanes for Connected and Automated Vehicles,* T2023/3, March 2023, TRAIL Thesis Series, the Netherlands

Eikenbroek, O., *Variations in Urban Traffic,* T2023/2, February 2023, TRAIL Thesis Series, the Netherlands

Wang, S., *Modeling Urban Automated Mobility on-Demand Systems: an Agent-Based Approach,* T2023/1, January 2023, TRAIL Thesis Series, the Netherlands

Szép, T., *Identifying Moral Antecedents of Decision-Making in Discrete Choice Models,* T2022/18, December 2022, TRAIL Thesis Series, the Netherlands

Zhou, Y., *Ship Behavior in Ports and Waterways: An empirical perspective,* T2022/17, December 2022, TRAIL Thesis Series, the Netherlands

Yan, Y., *Wear Behaviour of A Convex Pattern Surface for Bulk Handling Equipment,* T2022/16, December 2022, TRAIL Thesis Series, the Netherlands