Signal Group-based Look-ahead Traffic Signal Control

N.P.P. van Gurp

TU Delft

Goudappel Coffeng

Challenge the future

Signal Group-based Look-ahead Traffic Signal Control

by

N.P.P. van Gurp

to obtain the degree of Master of Science
at Delft University of Technology,
to be defended publicly on Thursday 31 January, 2019 at 15:00.

Student number:	4381947		
Project duration:	February 15, 2018 – January 31, 2019		
Thesis committee:	Dr. ir. A. Hegyi	TU Delft	(chairman)
	Prof. dr. ir. B. De Schutter	TU Delft	
	Dr. ir. A.M. Salomons	TU Delft	
	L. Krol MSc	Goudappel Coffeng	

This thesis is confidential and cannot be made public until August 1, 2019.

An electronic version of this thesis is available at http://repository.tudelft.nl/.

adviseurs mobiliteit
Goudappel Coffeng

TU Delft
Delft University of Technology

Preface

Whilst looking at the Rotterdam skyline from my studio flat window at some fifty meters of elevation, I look back at my TU Delft journey that started almost three-and-a-half years ago and is now about to end. In that time, I have gained a lot of knowledge, mastered a variety of skills, and met numerous new friends with whom I share many interests. The apotheosis of my education at TU Delft is this Master thesis, which I have really enjoyed working on over the past eleven months. I would like to thank the following people who have helped me in several ways to complete my Master thesis.

First of all, I would like to thank the members of my thesis committee. Andreas Hegyi and Maria Salomons, I really appreciate all of the questions that you have asked me throughout the process and all of the feedback you have provided me with, which surely has greatly contributed to the quality of this thesis. Bart De Schutter, thanks for your feedback and the ever quick response to various questions that I have asked you. Lieuwe Krol, thanks for your continuous support, for tirelessly answering all of my questions and for your help in making the traffic signal control algorithm work in Vissim. I have very much enjoyed working together with you.

Secondly, I would like to thank Goudappel Coffeng for offering me the opportunity to work on a project that perfectly fits my personal interests and that has intrigued me right from the beginning. I would also like to express my gratitude to all of my colleagues, in particular the people from the Traffic Management team in Deventer and the people from various other teams in The Hague, for offering me their support and creating a nice working environment. I look forward to starting my professional career as a traffic engineer at Goudappel Coffeng from February onward.

Thirdly, I would like to thank family and friends, of whom some I have met during my time in Delft, for their support. The many board game nights that we organised have allowed me to get my mind off work and relax, which contributed to a healthy work-life balance for most of the duration of the Master thesis project. Thanks also to Gerard te Vaanhold for joining me on a cold and rainy winter evening to make some great pictures of the N65 case study intersection, and for assisting me in configuring the thesis' cover.

Last but not least, I owe a debt of gratitude to my parents Elly and Wim, who have always believed in me, for their unconditional love and support.

N.P.P. van Gurp
Delft, January 2019

Summary

Introduction
Traffic signal controllers have been around since the late sixties of the 19th century and since then they have played an increasing role in traffic management. Thanks to developing technologies, the functionalities of traffic signal control algorithms have greatly enlarged over time. Current development of Dutch traffic signal control algorithms is based on the Talking Traffic program, which distinguishes three pillars:

- *Informing*; make data from the traffic signal controller available to road users (e.g. status of traffic signals (green/amber/red) and time-to-green and time-to-red predictions);
- *Prioritising*; favour certain road users over others (e.g. public transport) and communicate information on priorities to road users without using road-side equipment;
- *Optimising*; optimisation at both the intersection and network level by a different use of current data sources, by the use of additional data sources and by the use of new traffic signal control algorithms.

Traditional traffic signal control algorithms include inefficiencies in terms of these pillars. For example, the widely used fixed block structure may prevent the traffic signal controller from giving green to signal groups in the order in which the delay is minimised (optimisation pillar). Also, the variable green times that are applied to handle demand fluctuation (optimisation pillar) lead to an unpredictable cycle duration, which in turn leads to inaccurate time-to-green and time-to-red predictions (informing pillar).

With the introduction of the Golden Controller, a new traffic signal control algorithm by Goudappel Coffeng, a better performance is achieved for mainly the optimisation pillar. That is, by giving green to signal groups in a variable order, overall delay reduces. It, however, remains difficult to accurately predict control decisions and hence to align to all pillars at the same time.

To increase overall alignment, Goudappel Coffeng desires to develop a new traffic signal control algorithm that is able to create a planning of control decisions on the basis of expected vehicle arrivals. By doing so the predictability of control decisions increases while no sacrifices need to be made with regard to the optimality of the solution. This type of traffic signal control is referred to as look-ahead control.

Due to prediction errors such as inaccurate vehicle arrivals and imprecise queue discharge rates, and since priority requests may interfere with the order in which signal groups should be given green to minimise delay, full alignment with all pillars of Talking Traffic cannot be achieved. As a result, a balance needs to be found between *adaptability* of the planning to actual traffic conditions (i.e. delay reduction), *predictability* of control decisions and *priority* of road users. Goudappel Coffeng desires this balance to be user-adjustable, so that road authorities may decide themselves what objective(s) they find important.

Look-ahead Traffic Signal Control
In developing a new look-ahead traffic signal control algorithm, existing algorithms with similar functionalities are evaluated. The purpose of this evaluation is threefold and includes (a) obtaining general knowledge on look-ahead control, (b) preventing reinvention of the wheel and (c) locating design choices.

Traffic signal control by definition is characterised by a rolling horizon approach. In this approach control decisions are continuously made for a limited horizon during the entire interval in which the traffic signal controller is required to operate. In a rolling horizon approach, the horizon length and interval size are important design choices. Other design choices include recalculation frequency, prediction models (to estimate e.g. queueing) and the search technique. The latter one, which describes how the full set of possible control policies is searched through, is especially relevant since exponential growth of this set over the planning horizon does not allow for evaluation of all control policies within a reasonable amount of time.

It turns out that most of the existing algorithms that have been reviewed cannot easily be reused. This has to do with a computational heavy set-up of the algorithms and simplified approaches that are considered which do not support Dutch traffic signal control well. The algorithms by Sen and Head (1997), van Katwijk (2008) and Chen and Sun (2016), which rely on the same search technique but differ in the way in which signal groups are treated, are most relevant for this study.

Planning of Control Decisions (PCD) Algorithm
One of two parts of the new traffic signal control algorithm is the PCD algorithm. This algorithm uses queues at the moment of recalculation and expected vehicle arrivals to create a planning of control decisions. By using these inputs, the algorithm does not solely rely on (inaccurate) predicted information, but instead takes into account (accurate) actual traffic conditions too.

The PCD algorithm considers stages, in which signal groups can be present multiple times, instead of blocks. This increases the number of control options, which may lead to a better solution. Furthermore, the PCD algorithm applies a signal group-based approach over a stage-based approach, which allows signal groups to be given green ahead of time and which represents the behaviour of Dutch traffic signal controllers well. This, in turn, is beneficial for the accuracy of time-to-green and time-to-red predictions. Priority is interwoven in the PCD algorithm. That is, by considering the composition of a queue and the weight of each of its components any form of priority can be given to road users on the level of signal groups, vehicles classes and even individual vehicles.

Like the three earlier mentioned algorithms (Sen and Head (1997), van Katwijk (2008) and Chen and Sun (2016)), the PCD algorithm is a forward recursion dynamic programming formulation. This formulation prevents that the full set of possible control policies has to be searched, which allows for a great reduction of the computation time. Despite this efficient search technique, computation times are still rather high.

Traffic Signal Control (TSC) Algorithm
The TSC algorithm is the second part of the new traffic signal control algorithm. The TSC algorithm uses the planning of control decisions from the PCD algorithm as a framework to grant the right on green to signal groups. The TSC algorithm basically adapts the planning to actual traffic conditions and it determines the colour of all signals at the intersection at every time instant. Although the TSC algorithm incorporates various concepts of standard Dutch traffic signal control (e.g. signal completion, the usage of loop detectors to extend and end green), many of the algorithm's components are newly developed in this study. None of the authors of existing look-ahead traffic signal control algorithms namely elaborate on how to adapt a static planning of control decisions to actual traffic conditions.

The balance between adaptability and predictability is included in the TSC algorithm. That is, next to green extension via loop detectors instead of strictly copying the planning, a parameter is specified that allows signal groups to be green for some time longer than the green time that is already allocated to these signal groups in the planning of control decisions.

Algorithm Evaluation
Using the N65 intersection near the Dutch town of Helvoirt as a case study, the performance of the new traffic signal control algorithm is compared to the performance of other algorithms. The new algorithm is compared to the traditional actuated traffic signal control algorithm [1] and the Golden Controller [2]. Both algorithms have been developed by Goudappel Coffeng. Delay and queue length are considered as performance indicators, as well as the time-to-green and time-to-red prediction accuracy. The prediction accuracies are not compared to [1] and [2], because unlike the new algorithm these algorithms do not include the ability to present time-to-green and time-to-red predictions to road users.

In terms of delay, the new algorithm outperforms [1]. That is, if the delay of [1] is considered 100%, then the new algorithm achieves a delay of only 96.9% to 95.3%, depending on traffic flows and the availability of vehicle arrivals. The new algorithm does not yet achieve the same delay reduction that [2] achieves (91.5% to 90.9%). Further perfecting of the new algorithm is required to increase delay reduction.

A more detailed look into the delays for vehicles at individual signal groups reveals that the new algorithm further increases inequality at the intersection. That is, signal groups with high traffic flows are given even more green than [2] does, which results in further reduction of the delay for vehicles at these signal groups, and which is at the expense of the average delay of vehicles at other signal groups. The 95$^{\text{th}}$ percentile of the queue length supports this observation.

With respect to time-to-green and time-to-red, predictions are fairly accurate for most signal groups. That is, the average inaccuracy equals approximately -0.82 seconds, which indicates that the time that is required to clear a queue is slightly underestimated (this could be solved by setting different values for the user-adjustable queue discharge rates), and the average standard deviation equals some 2 seconds for time-to-green and some 4 seconds for time-to-red predictions.

Conclusions and Future Work
In conclusion, although the new traffic signal control algorithm that is developed in this study is certainly not perfect yet, its potential is clear. A wide range of improvement possibilities are suggested for both the PCD and TSC algorithms to further reduce delay and to increase the user-adjustable balance between the different objectives. Without a doubt, the computation time of the PCD algorithm is the main limitation that prevents direct implementation of the new traffic signal control algorithm on the street. Although various conceptual solutions are named to reduce computation time of the PCD algorithm, the magnitude of the computation times will most likely require parallelisation of the PCD and TSC algorithms. It is currently unclear how this parallelisation should be implemented.

Contents

1

Introduction

Since their first implementation in London's Westminster district in 1868 (Londonist, 2017), traffic signal controllers have played a major role in (urban) traffic management all around the world. Over time, increasing computing power, enhanced possibilities for detection of vehicles and improved communication technologies have allowed for sophisticated traffic signal control algorithms that include all kinds of features, and development continues to go on. In this thesis a new traffic signal control algorithm is developed and this first chapter describes why and the context wherein this is done. Before continuing, the reader is advised to read Appendix A if he or she is unfamiliar with the basics of traffic signal control.

Section 1.1 begins the chapter by briefly introducing Talking Traffic. Talking Traffic is an umbrella term for all kinds of technologies that involve communication between vehicles and roadside equipment. Talking Traffic is at the foundation of present day traffic signal control development and as a matter of fact the Golden Controller also relies hereupon. The Golden Controller, which is described in Section 1.2, is a new type of traffic signal control algorithm that has recently been developed by Goudappel Coffeng. The Golden Controller is able to further reduce delays in comparison to traditional traffic signal control algorithms. Goudappel Coffeng desires to develop a new traffic signal control algorithm that adopts the basic principles and strengths of the Golden Controller, but also adheres to the concepts that Section 1.3 describes. In Section 1.4 the goal and research questions are described and Section 1.5 further defines and delimits the research. Finally, Section 1.6 presents the outline of this thesis.

1.1. Talking Traffic

With the universal rise of global positioning systems and the increased accessibility of internet, the door is opened for new technologies. In traffic, these new technologies come in the form of intelligent transportation systems (ITS). Within ITS, connected vehicle technology, which includes wireless vehicle to vehicle (V2V), vehicle to infrastructure (V2I) and infrastructure to vehicle (I2V) communication, is among the most promising. Thanks to the availability of detailed data such as the position, speed and acceleration of vehicles (Chandan et al., 2017), connected vehicle technology is thought to significantly improve the operation of traffic at signalised intersections (Guler et al., 2014; Goudappel Coffeng, 2016b). It is hence not surprising that present day traffic signal control development relies on this technology.

In the Netherlands connected vehicle technology is caught in the Talking Traffic partnership. Talking Traffic is introduced in the Beter Benutten program, which is run as a collaboration between the Dutch government and a number of private corporations (Platform Beter Benutten, n.d.; van Katwijk et al., 2016). The traffic signal control part of Talking Traffic distinguishes three pillars to focus on for further development of traffic signal control algorithms (Trafficquest, 2017; Goudappel Coffeng, 2016c):

- *Informing*; make data from the traffic signal controller available to road users. Among this data are the status of traffic signals (green/amber/red) and speed advice on the basis of time-to-green and time-to-red predictions[1];

[1] The definition of time-to-green and time-to-red can be found in Appendix B. This appendix also includes definitions of many other traffic signal control terms, as well as the corresponding English-Dutch translations.

- *Prioritising*; favour certain road users over others (e.g. public transport) and communicate information on priorities to road users without using road-side equipment;
- *Optimising*; optimisation at both the intersection and network level by a different use of current data sources, by the use of additional data sources and by the use of new traffic signal control algorithms.

Traditional traffic signal control strategies have difficulty aligning to all three pillars at the same time. For instance, the variable green times that are required to handle demand fluctuation (optimisation pillar) lead to an unpredictable cycle duration and as a result it is difficult to provide traffic with reliable time-to-green and time-to-red predictions[2] (informing pillar). There is also certain inefficiency in terms of individual pillars. The fixed block structure that is found in many traditional traffic signal control algorithms prevents the traffic signal controller from giving green to the combination of signal groups that together reduce the delay most (optimisation pillar). These signal groups may namely either not form a block together, or the fixed order of blocks allows other signal groups to become green first.

1.2. Golden Controller

With the development of the Golden Controller, Goudappel Coffeng tries to better align to the pillars of Talking Traffic[3] and at the same time she tries to counter the above mentioned limitations of traditional traffic signal control algorithms. The Golden Controller is a project in development and consists of five so called modules: *Golden Green*, *Golden Change*, *Golden Choice*, *Golden Map* and *Golden Info*. Of all modules the first three mentioned are currently operational and the other two modules are under development. The next paragraphs provide a brief explanation of the modules.

Golden Green
In the Golden Green module the maximum extension green time of signal groups is based on the estimated number of vehicles in a queue, rather than on statistics like traditional traffic signal controllers do (Wilson, 2014). Using this approach, the Golden Controller prevents inefficient extension of green[4] and as a result the degree of green utilisation increases. For the purpose of counting vehicles, the stop line and distant loop detectors are used. If a queue grows upstream of the distant loop detector or when a distant loop detector is not present, vehicles are added on the basis of the estimated traffic flow.

Golden Change
Within the Golden Change module signal groups that may receive green simultaneously are defined manually and stored as stages. The Golden Change module continuously determines the cumulative delay for each stage on the basis of the estimated number of vehicles in the queues and it grants the right on green to the signal groups of the stage with the largest cumulative delay. By using such an algorithm, there is no longer a fixed order in which signal groups receive green and green can be given to the signal groups of the stage with the highest delay.

Golden Choice
The Golden Choice module is meant for traffic networks (i.e. coordination of multiple traffic signal controllers). Instead of only communicating traffic flows between the traffic signal controllers, the Golden Choice module also transfers whether or not vehicles have remained in the queue after green has ended (in Dutch: 'overstaan'), which allows for a better choice between signal control options as there is more information available.

Golden Map
The Golden Map, which is currently under development, is intended to be a module that tracks the locations of vehicles and holds additional data of these vehicles in terms of their speed and their type. In the Golden Map module vehicles are tracked on the basis of loop detector data. That is, detector occupancy is translated

[2] Predicting time-to-green and time-to-red times is possible on the basis of previous green durations, however, especially for off-peak periods predictions are fairly inaccurate (Goudappel Coffeng, 2016a).
[3] Although Goudappel Coffeng does not participate in the Talking Traffic program, Goudappel Coffeng finds it important to contribute to traffic signal control development and to stay an important player within the software side of the (Dutch) traffic signal control market accordingly.
[4] Inefficient extension of green occurs when green is extended for a few vehicles only that arrive at an intersection's signal group consecutively and that all fall just within the gap time of the loop detectors.

into vehicle presence and these vehicles are then extrapolated in time and space using a car following model[5] (Wiedemann '74). The Golden Map is intended to track vehicles on the basis of floating car data[6] in the future.

Golden Info
The Golden Info module is meant to determine and present time-to-green and time-to-red predictions to road users. The desire to include these predictions in a traffic signal controller originates from a number of positive effects. Islam et al. (2016) found that there is a small reduction in start-up lost time at signalised intersections with a red signal countdown timer and it was also found that drivers show safer responses at intersections with green signal countdown timers (Islam et al., 2017). Furthermore, it is thought that road users perceive less waiting time when they know their waiting time beforehand (van der Bijl et al., 2011). Moreover, providing road users with time-to-green predictions has the potential to reduce emissions. There currently exists a basic version of the Golden Info module, but further development is required.

On March 27, 2018 the Golden Green and Golden Change modules of the Golden Controller made their first appearance on an N65 intersection near the Dutch town of Helvoirt.

1.3. Desired New Features

Although a recent study by Goudappel Coffeng (2019) shows that the Golden Controller is able to reduce delays in comparison to standard Dutch actuated traffic signal control (optimisation pillar), the control decisions remain fairly unpredictable and hence it is still difficult to provide road users with reliable time-to-green and time-to-red predictions (informing pillar). Goudappel Coffeng desires to develop a new traffic signal control algorithm that better aligns with all pillars of Talking Traffic by taking into account the two features that are described in Sections 1.3.1 and 1.3.2.

1.3.1. Look-ahead Control

Goudappel Coffeng would like to increase the predictability of control decisions by including short-term predictions of vehicle arrivals in the new traffic signal control algorithm. The idea is that information on the number and type of vehicles that will arrive at the intersection in the near future, allows the traffic signal controller to determine when and how long signal groups should be given green and consequently a short-term planning of control decisions can be determined. Establishing a planning in advance of the arrival of vehicles allows for more reliable time-to-green and time-to-red predictions, while no sacrifices need to be made in terms of performance of the solution. The dynamic sequence of stages that allows the Golden Controller to reduce delays namely remains available.

Next to improving the accuracy of time-to-green and time-to-red predictions, it is thought that basing control decisions on predicted vehicle arrivals will also further reduce delays. The idea behind this is as follows: if a large platoon of vehicles is detected upstream of the traffic signal controller, the traffic signal controller could anticipate to that in a way that these vehicles are able to pass through the intersection (relatively) freely at the expense of a reasonably prolonged delay of other vehicles at the intersection. Jiang et al. (2006) show that this is especially useful for major-minor type of intersections, such as the N65 intersection near Helvoirt.

1.3.2. Objective Balancement

Although adding short-term predicted vehicle arrivals will likely allow for more reliable time-to-green and time-to-red predictions, full adaptability and a high predictability of control decisions remain contradictory objectives. That is, the more reliable the time-to-green and time-to-red predictions, the less the control decisions may be exposed to last-minute changes. A similar story of conflicting objectives holds in case of prioritising certain road users over others. The road user that is to be prioritised may force the traffic signal controller to give green to signal groups in an order in which delay reduction is non-optimal.

Prediction errors such as vehicles not being spotted in time (this is especially relevant for urban areas) and mismatches between the actual and the modelled behaviour of vehicles cause differences between predicted and actual vehicle arrivals. To deal with these differences, the traffic signal controller might want to deviate

[5] A car following model describes the lateral action of a vehicle in terms of its position, speed or acceleration as a function the vehicle's leading vehicle(s) (Knoop, 2017).

[6] Floating car data originate from devices that are equipped with a global positioning system and that travel along with vehicles. Example devices are the mobile phones of vehicle occupants and navigation systems. Traffic conditions (e.g. flow and speed) can be derived from this data (Felici, 2017; Trafficquest, 2017).

from its original planning, for instance by changing the order in which signal groups receive green or by extending green times. Although traffic may benefit from these last minute changes in terms of delay reduction, a negative effect occurs for the accuracy of predicted time-to-green and time-to-red times. As long as prediction errors continue to exist, full alignment with all pillars of Talking Traffic cannot be achieved. As a result, a balance needs to be found between delay minimisation (which could be viewed as adaptability of the planning to actual traffic conditions), predictability of control decisions and priority of road users. Goudappel Coffeng desires this balance to be user-adjustable, so that road authorities are able to choose their optimal balance between the different objectives.

1.4. Goal and Research Questions

Concluding, two features define a new traffic signal control algorithm which is inspired by the Golden Controller. First, reduce delays and improve the reliability of time-to-green and time-to-red predictions by making a short-term planning of control decisions on the basis of predicted vehicle arrivals. Secondly, within this new traffic signal control algorithm allow road authorities to choose their optimal balance between the different objectives of adaptability, predictability and priority. The research goal is described as follows:

> *The goal of the research is to develop a new traffic signal control algorithm that relies on predicted vehicle arrivals to make a short-term planning of control decisions, that adjusts these control decisions on the basis of real time traffic conditions (adaptability) and that thereby takes into account the user-adjustable objectives of predictability of control decisions and priority of road users.*

Beside the aspects that are mentioned in the research goal, there is another aspect that plays a role in the development of the new traffic signal control algorithm. Since the new algorithm is required to run not only once but instead in recurring manner (see Section 2.1), computational efficiency is a topic that is to be considered. Also, development of a new algorithm only makes sense if the algorithm is evaluated for its performance thereafter and hence this is the final part of the research.

Given the complexity of the research goal, a single research question does not do. The next research questions, that each require some topics to be examined, are to be answered in this study:

- What is the definition of a vehicle arrival, which data is required from vehicles, and how can vehicle arrivals be determined from this data?
- How is a short-term planning of control decisions created and are there any existing algorithms that can, whether or not after modification, be reused for this purpose?
- How can the objectives of predictability and priority be made user-adjustable and how can they, together with adaptability, be balanced within the new traffic signal control algorithm?
- How is computational efficiency of the new algorithm defined and achieved?
- How does the new algorithm perform in comparison to other traffic signal control algorithms?

To answer the research questions mentioned above, different methods are considered. The first and third questions are clear examples of design problems, for which the answers will become clear as work on this thesis progresses. The answers to the second and fourth questions fully or mainly rely on the literature review that is presented in the next chapter. The fifth research question is answered by means of a simulation study.

1.5. Scope of the Study

By presenting some a priori considerations, the next three paragraphs further define and delimit the research.

Algorithm Application

The new traffic signal control algorithm will be developed for single intersections only. The N65 intersection near Helvoirt, a standard four-legged Dutch intersection with exclusive signal groups for cyclists (no pedestrians), will play the role of case study. Figure 1.1 presents an aerial view of this intersection.

The actual traffic system is sometimes more comprehensive than single intersections only. It is, however, the specific desire of Goudappel Coffeng to not develop a traffic signal control algorithm that is even more complex and more of a black box than existing algorithms already are that determine an optimal solution over a network of intersections. The new algorithm has yet to keep in mind its position in a network, which is a topic that is discussed in Chapter 6.

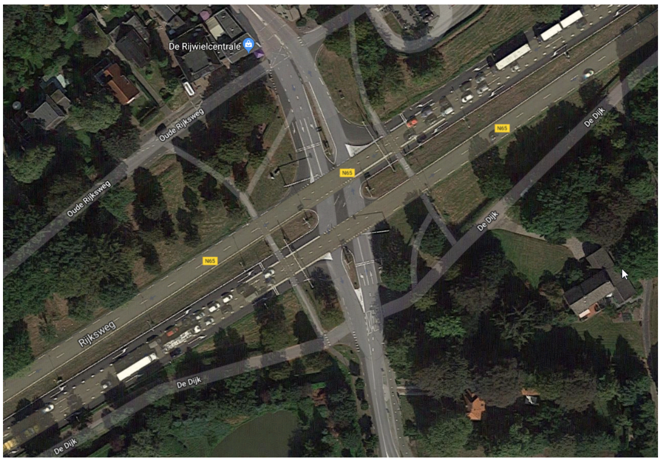


Figure 1.1: Aerial view of the N65 case study (figure source: Google, 2018).

Vehicle Arrival Predictions

The key to making a good planning is that information on the basis of which the planning is determined is available for the full horizon of the planning. For this study this means that the traffic signal controller is required to know when vehicles are expected to arrive at the intersection tens of seconds before they actually arrive. Goudappel Coffeng ideally wants to obtain these vehicle arrivals via the Golden Map module (see Section 1.2), however, this module is not used in this study. Two arguments underlie this decision.

First, since the Golden Map module has not yet been completed and evaluated, extensive calibration and validation is required. Not only does this take a lot of time, but it also shifts the attention away from the core purpose of this study, which is developing a new traffic signal control algorithm[7].

Secondly, the mathematical model that the Golden Map uses for extrapolation of vehicle locations very likely results in a mismatch of unknown scale between predicted and actual vehicle arrivals. This does not fit the scientific desire to have a controlled experiment in which a self-defined error can be introduced.

As a result, alternative approaches to predict vehicle arrivals are to be considered. Within this context one could think of a number of approaches. In this thesis the *Run and Repeat* approach that is explained in Explanation 1.1 is considered.

Driving Behaviour Changes

The algorithm that is to be developed will present time-to-green and time-to-red predictions to road users. As a result of these predictions, driving behaviour of human drivers might change (e.g. increased acceleration if green is about to terminate). Since many microsimulation models do not (yet) incorporate time-to-green and time-to-red predictions and since the impact of these predictions on traffic flow is only small (see Section 1.2), driving behaviour changes are considered irrelevant for this study and they are hence out of scope.

1.6. Thesis Outline

The remainder of this thesis is organised as follows.

In Chapter 2 literature on look-ahead traffic signal control is reviewed. This chapter explores existing look-ahead traffic signal control algorithms for their techniques to determine a short-term planning of control

[7] The development and calibration/validation of such prediction models is a topic in which many hours can be spend as Chandan et al. (2017) suggest who have tried to estimate the locations of vehicles on the basis of Wiedemann '74 car following model. One could even take it to the level of including artificial intelligence and self-learning to improve the accuracy of vehicle arrival predictions. A study by Helmy (2017) in which a multi-task learning recurrent neural network with exogenous inputs (NARX) for short-term predictions of vehicle arrivals has been developed, could be of use in this context.

decisions and describes whether and how parts of these algorithms can be (modified and) reused. This chapter also identifies where design choices are located within these algorithms.

Chapter 3 is dedicated to the planning of control decisions (PCD) algorithm. On the basis of, among other things, queues at the moment of recalculation and expected vehicle arrivals this algorithm determines the order in which signal groups should be given green and the corresponding green duration so that the fit with a control objective is optimised. This can then be translated into time-to-green and time-to-red predictions.

In Chapter 4 the traffic signal control (TSC) algorithm is described. This algorithm uses the output from the PCD algorithm as a framework to grant the right on green to signal groups and it thereby basically determines the colours of all of signals for every time instant. In this algorithm adaptation of the planning of control decisions to actual traffic conditions takes place.

Chapter 5 evaluates the performance of the new traffic signal control algorithm (the PCD and TSC algorithms combined) in terms of a set of performance indicators and on the basis of various scenarios. The new algorithm is evaluated in a qualitative and quantitative manner.

Finally, Chapter 6 concludes the results of the research. In this chapter future work is also discussed.

Explanation 1.1: Run and Repeat approach

The following series of actions defines the Run and Repeat approach. First, pick a traffic signal control algorithm and apply it to a signalised intersection in a microsimulation model. Next, select a random seed and run the simulation model once to obtain data on the location of vehicles. Then, translate this data into vehicle arrivals (see Section 3.2). Finally, run the exact same simulation run again (i.e. same random seed) and use the vehicle arrivals that were determined earlier as an input for the new traffic signal control algorithm.

The main advantage of this approach is that data on the location of vehicles in the first simulation run largely represents the actual location of vehicles in the second simulation run. After all, the way in which vehicles move towards the intersection does not differ between the model runs. Other approaches that rely heavier on extrapolation are likely to produce larger differences between the predicted and actual location of vehicles.

Since the control decisions that were made during the first simulation run are very likely not the exact same control decisions that the new traffic signal control algorithm makes in the second simulation run, differences in queue lengths arise. As a result, one cannot simply copy all of the locational data of vehicles from the first simulation run to the second simulation run. By defining a parameter called the *accuracy distance* one can work around this (see Figure 1.2).

The accuracy distance denotes the point in space for which the user considers locational data of the first simulation run to be sufficiently accurate to be used in the second simulation run. Ideally the accuracy distance is set to a low value. A low value namely reduces the share of linear extrapolation in predicting the location of vehicles, thereby limiting the prediction error. The accuracy distance can, however, not be too low either, because of the increasing influence of queues that arose during the first simulation run.

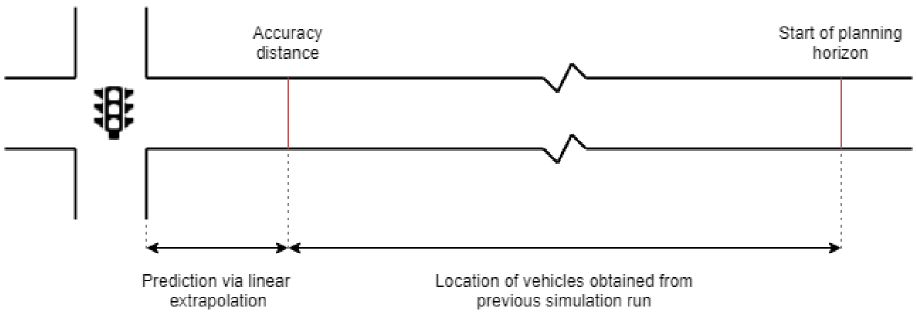


Figure 1.2: Visual representation of the Run and Repeat approach. Between the start of the planning horizon and the accuracy distance the location of vehicles is obtained from the first simulation run. When a vehicle reaches the accuracy distance, prediction of its location takes place via linear extrapolation.

2

Look-ahead Traffic Signal Control

Establishing a short-term planning of control decisions on the basis of predicted vehicle arrivals, or in short look-ahead control, is not a new idea. Instead, already in the seventies Robertson and Bretherton (1974) presented one of the first algorithms of this kind and the volume of scientific work on look-ahead control has increased ever since. In this chapter international scientific literature is explored with the threefold goal of (a) obtaining general knowledge on look-ahead control, (b) assessing whether or not parts of existing algorithms can be reused (to prevent reinvention of the wheel) and (c) locating design choices and determining their implications. Exploration of literature is performed on the basis of four key design choices that van Katwijk (2008) defines for look-ahead traffic signal control:

- The *planning horizon* describes the number of intervals and interval size (resolution) over which the control policy is determined. The number of intervals may be fixed or may depend on traffic conditions and the resolution may be static or dynamic;
- The *update frequency* relates to how often recalculations are performed. Optimisation could take place during static intervals (e.g. ten times per second) or it could take place dynamically (e.g. when green of a signal group ends);
- The *search algorithm* represents the technique that is used to search the full set of control options and find the sequence of control decisions that fits a control objective best. This aspects is closely related to the computation time of the algorithm;
- The traffic signal control algorithm requires *prediction models* to predict vehicle arrivals and evaluate the performance of control decisions in terms of e.g. delay and queueing.

Each of the sections of this chapter elaborates on one or more of the above mentioned design choices. In Section 2.1 the planning horizon and update frequency are discussed, Section 2.2 is dedicated to search algorithms and prediction models are the topic of Section 2.3. Section 2.4 provides a summary of the chapter.

2.1. Planning Horizon and Update Frequency

In theory traffic signal controllers are able to operate for an infinite amount of time. Since traffic data such as loop detector occupancies and predicted vehicle arrivals are only available a short while before a control decision is to be made, single time execution of the optimisation algorithm is impossible. This statement is emphasised by the extensive computational load that single time execution over a very large time horizon would cause (see Section 2.2 for further explanation). As a result, by definition traffic signal control algorithms require some sort of approach where control decisions are continuously made for a limited horizon during the entire interval in which the traffic signal controller is required to operate. Porche and Lafortune (1997) refer to such an approach as *rolling horizon* and herein the time horizon continues to roll forward each time the traffic signal control algorithm makes a control decision.

Robertson and Bretherton (1974) and Zheng and Recker (2013) introduce the rolling horizon concept as follows. First, a planning horizon is determined that consists of N time intervals. The entire horizon is then split up into two parts. The head part H includes a time period over which detected or actual traffic information is available. This part consists of one or multiple intervals and could be as small as a tenth of a second. The tail part $N-H$ includes a time period over which only predicted traffic information is available. On the basis of an objective function that often consists of minimising delays and by using some sort of search algorithm, a sequence of control decisions is determined that forms the control policy. These control decisions are determined for the entire planning horizon $H+(N-H)=N$, but they are only implemented for the head part H. Following the execution of the first control decision, the planning horizon is shifted into the future over R intervals (roll period, often equal to H), after which the process is repeated with new information that has come available. A scheme of the rolling horizon approach is presented in Figure 2.1.

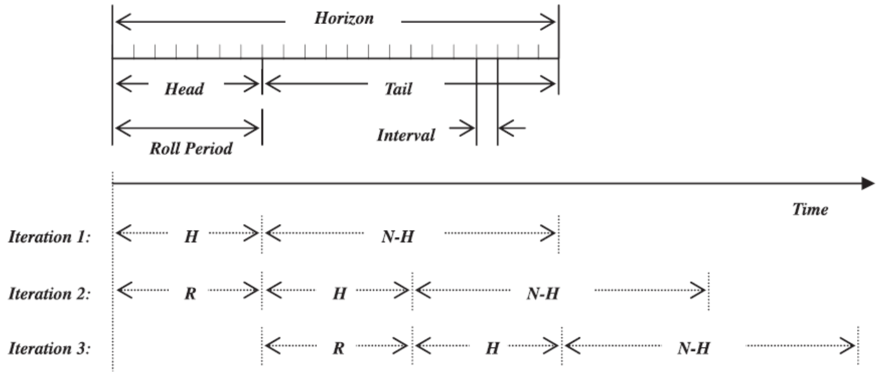


Figure 2.1: A rolling horizon approach for traffic signal control (figure source: Zheng and Recker, 2013).

Although most existing algorithms apply a rolling horizon approach wherein the number of intervals N as well as the interval size is fixed, some authors consider different approaches.

Porche and Lafortune (1997) make the horizon depend on the traffic situation. That is, at the time of recalculation the algorithm considers all current vehicles and expected vehicle arrivals within the horizon and control decisions are determined for as long as there are vehicles that have not passed the intersection.

Zheng and Recker (2013) wields the North American NEMA standard (see Explanation 2.1) as a starting point and sets the head and tail parts equal to two consecutive barriers groups or one full cycle. After the control policy is determined, the horizon is shifted one barrier group into the future.

In the above approaches the end of the horizon is picked wisely and calculation of the control policy is performed over a horizon of interest only. The usefulness of these approaches for the new algorithm is however limited. In the first approach the horizon may become rather large and the second approach is very much NEMA based, which does not suit Dutch practice. Choosing the right moment to recalculate so that unnecessary recalculations are prevented, which often but not necessarily implies the application of a variable update frequency, is a more feasible option to relax the algorithm's computational requirements.

A variable resolution as proposed by van Katwijk (2008) could be another way to reduce computational load. By increasing the interval size as the planning progresses into the future (see Figure 2.2), the length of the horizon increases while the number of intervals (and thus control options) remains constant. This then either allows for increased time-to-green and time-to-red predictions without having to incorporate longer computation times, or a decreased computation time if the horizon length is kept equal (less intervals).

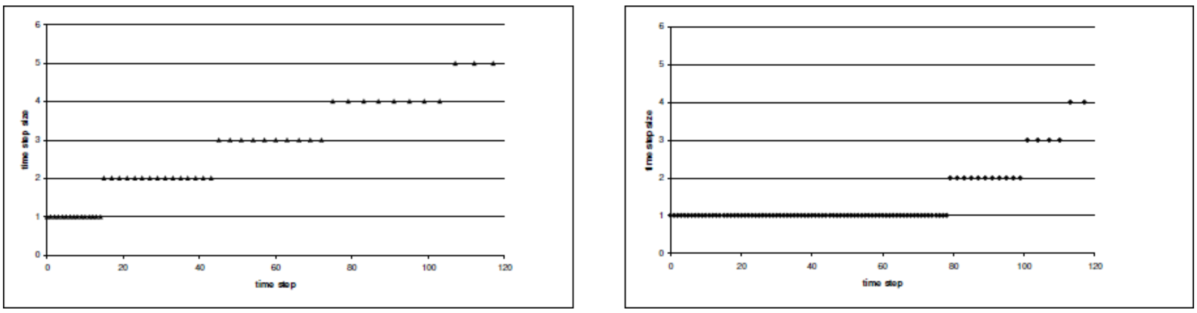


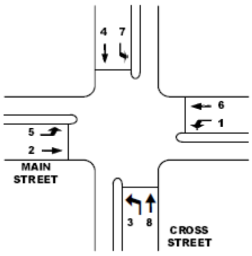
Figure 2.2: Two planning horizons that consider different resolutions. A linearly increasing interval size is presented on the left, and an exponentially increasing interval size is presented on the right (figure source: van Katwijk, 2008).

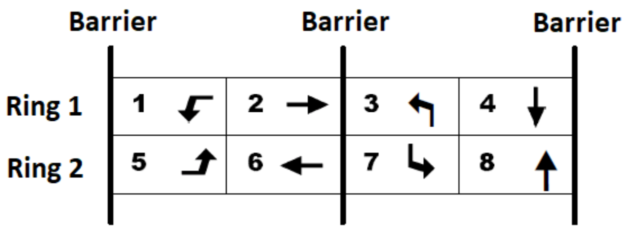
Several authors suggest that, independent of going with either a fixed or variable number of intervals, choosing the right length of the horizon is important. Both a too long and a too short horizon will namely prevent well functioning of the new algorithm:

- In case of a *too long* planning horizon, attention is shifted away from the head part to the tail part. Not only is the tail part not served in the current time instant, but control decisions for this part may never be implemented either (Cai et al., 2009). Furthermore, picking a too long tail shifts weight from accurate detected traffic data to less accurate predicted data and computation time, which is scarce in a rolling horizon approach anyway, is unnecessarily increased. In addition, Robertson and Bretherton (1974) argue that optimality in control is not particularly sensitive to variations in traffic arrivals beyond 25 seconds in the future;
- A *too short* planning horizon leads to short-sighted control decisions which in turn may lead to non-optimality. Also, a short planning horizon restricts the freedom of the control decisions that can be taken. For instance, if a planning horizon of 5-10 seconds is considered, then the traffic signal controller may only decide on extending or ending green. If a planning horizon is considered of e.g. 60 seconds, then the traffic signal controller may decide on the sequencing of signal groups for their green time too.

Explanation 2.1: NEMA Standard for Traffic Signal Control in North America

NEMA, an abbreviation for National Electrical Manufacturers Association, is an organisation that has developed the North American standard for traffic signal control. The NEMA standard includes a ring-barrier structure in which eight signal groups (called 'phases' in American literature) are defined for a typical intersection. A ring corresponds to a set of self-conflicting phases and barriers limit the side of the intersection that is served (i.e. north-south or east-west). Main street phases are typically numbered 1, 2, 5 and 6 and side street phases are numbered 3, 4, 7 and 8. The figures below present a typical American intersection and its corresponding NEMA ring-barrier structure.





The idea of the ring-barrier structure is that there is always one phase active from either rings and both of the active phases must be located within the same barriers. In total this leaves four phase combinations per barrier group (e.g. 1+5, 1+6, 2+5, 2+6), or eight phase combinations for the entire intersection. The traffic signal controller may only pass a barrier after a user-defined all-red time has elapsed during which all signals at the intersection show red (Rodegerdts et al., 2004).

Compared to the Dutch approach of traffic signal control (see Appendix A), NEMA is incomplete and inefficient. That is, right turning signal groups and cyclist/pedestrians are missing[a] and a single all-red time value is less efficient when compared to the signal group-based clearance times in the Dutch approach. One could try to complement NEMA for the Dutch situation, however, the flexibility that the Dutch approach offers with primary ahead and alternative realisations cannot be achieved[b]. The applicability of NEMA for Dutch signalised intersections is thus limited.

[a] Many American intersections have dedicated lanes for right turning signal groups that fall outside of the intersection and that hence do not require signalling. In case dedicated lanes are absent, then traffic is allowed by law to turn right on a red light in most states in the USA, again omitting the need for devotion of dedicated phases. In case cyclists and pedestrians are present on the intersection, they are often given green simultaneous with the corresponding through going phase as a permitted conflict.

[b] For example, let 1 and 6 be the phases that currently receive green and let there be some pedestrian and cyclist signal group parallel and incorporated into phase 4. If there is no more traffic for phase 6 then that phase may end and the pedestrians and cyclists of phase 4 may receive green. Due to phase 1 still receiving green, motorised vehicles of phase 4 may not receive green. Since phase 4 as a whole cannot be given green, green is not realised for the cyclists and pedestrians either. Also, phase 4 is in another barrier group than phase 1 and hence phase 4 is not allowed to become green.

2.2. Search Techniques

Look-ahead control essentially comes down to finding the one sequence of control decisions that fits the control objective best within an exhausting set of possibilities. The full set of control decisions is called the decision space and it is typically viewed as a decision tree (see Figure 2.3). As can be observed from the figure, the decision space grows exponentially with the number of intervals. Due to the limited amount of time that is available for recalculation of the control policy in a rolling horizon approach, brute force enumeration and evaluation of all sequences of control decisions will not do. Computational efficiency, which can be defined as the degree to which the entire decision tree will have to be built to find the optimal control sequence (Porche and Lafortune, 1997), is hence an important topic in developing the new algorithm. By stating that the computation time is one of the fundamental limitations in developing traffic signal control algorithms, Zheng and Recker (2013) emphasise the topic's importance.

This section is dedicated to search techniques, i.e. the way in which a decision tree is built and searched through, and it starts by describing a classification of search techniques in Section 2.2.1. Section 2.2.2 explores existing look-ahead algorithms for their search techniques and in Section 2.2.3 dynamic programming, a search technique of which it turns out it is commonly applied in look-ahead control, is elaborated on.

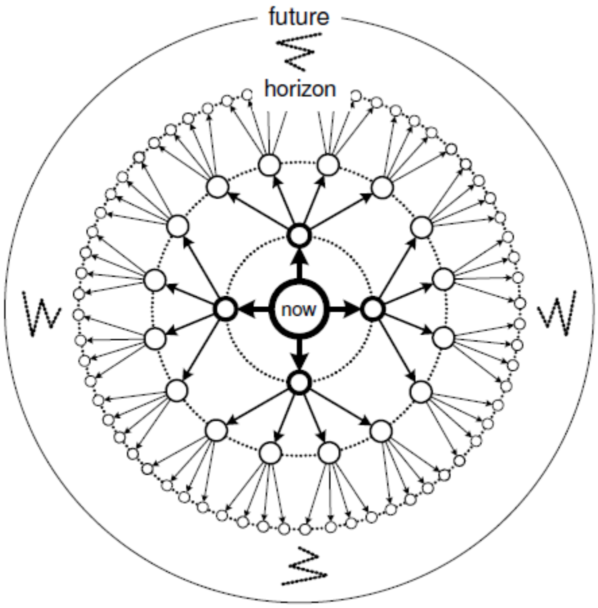


Figure 2.3: An example of a decision tree. All of the circles in the figure represent control decisions (figure source: van Katwijk, 2008).

2.2.1. Classification of Search Techniques

Search algorithms are typically distinguished on the basis of two characteristics (van Katwijk, 2008):

- *Complete* versus *incomplete* exploration; algorithms with complete exploration investigate the entire decision space and as a result these search algorithms will eventually always come up with the optimal control sequence. Search algorithms based on incomplete exploration do not explore the entire decision space and hence they cannot guarantee that the optimal control sequence is found. Incomplete exploration is useful when a relatively good solution is sufficient, since not having to explore all possible control sequences saves a lot of time, especially in case of large decision spaces;
- *Move-based* versus *constructive* methods; move-based search methods repeatedly move to adjacent solutions given a candidate solution as a starting-point (see Figure 2.4). Move-based methods only investigate the neighbourhood of the candidate solution and they are often referred to as local-search methods. Move-based methods are typically incomplete and they aim to find the local optimal solution. Move-based methods cannot guarantee that the global optimal solution is found. After all, the local optimum found by the algorithm could by chance be the global optimum, but it could just as well be that the global optimum lies outside of the neighbourhood that is explored. Two examples of move-based methods are hill-climbing and genetic algorithms.

Constructive search methods organise the decision space by partitioning it systematically using, e.g., a tree structure that allows for systematical (and complete) traversing through the decision space. Contrary to move-based methods, constructive methods are able to exclude large subspaces a priori since they allow for reasoning about whole classes of similar solutions. For example, the algorithm can decide to skip exploration of a part of the decision tree if it is able to determine beforehand that this part will not hold the optimal control policy. Constructive search methods that are typically used in traffic signal control are branch-and-bound and dynamic programming.

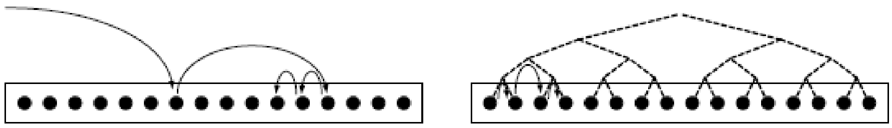


Figure 2.4: The left figure shows a set of solutions (black dots) that is explored using a move-based method. The black arrows mark the way in which solutions are traversed and the lack of a neat structure herein is clearly visible. In the right figure the set of solutions is explored using a constructive search algorithm. Here there is a clear order in which solutions are traversed (figure source: van Katwijk, 2008).

Although in theory traffic signal control lends itself for all of the different search methods mentioned above, Section 2.2.2 reveals that constructive search methods - and in particular dynamic programming - are often found in look-ahead traffic signal control. Cai et al. (2009) even mention that dynamic programming is the only technique that is able to generate an exact solution for optimisation over time.

2.2.2. Search Techniques in Existing Algorithms

The authors of a number of scientific works that have been reviewed either do not specifically mention how the decision space is searched through (e.g. Gradinescu et al. (2007) and Kari et al. (2014)) or the full set of control options is manually defined and looped through, which is only possible because of the limited number of control options (e.g. Zheng and Recker (2013)). Authors that do elaborate on the search techniques in their algorithms reveal that the variety herein is rather limited. He et al. (2012) apply mixed-integer linear programming, however, the authors mention that large mixed-integer linear programming formulations cannot easily be solved real-time. To address this problem the authors aggregate individual vehicles into platoons, which does not necessarily comply with the level of detail that Goudappel Coffeng has in mind. Pandit et al. (2013) apply another approach, which is job scheduling with conflicts. Most algorithms, however, apply dynamic programming (e.g. Henry et al. (1983), Mirchandani and Head (2001) and Feng et al. (2015)). In some other works dynamic programming is combined with other techniques such as complete enumeration (e.g. Priemer and Friedrich (2009)) and optimal sequential search (e.g. Gartner (1983)).

2.2.3. Dynamic Programming

Dynamic programming is a search technique that has been developed by Bellman (1957). In dynamic programming a problem is decomposed into a number of subproblems and the solutions to these subproblems are stored so that they can be reused for the final solution. The key feature of dynamic programming is that time complexity of algorithms can be reduced from exponential to polynomial level, which is hugely beneficial for the computation time of an algorithm (Wu, 2017). According to Demaine (2011), an optimisation problem to which dynamic programming can be applied includes two properties:

- *Overlapping subproblems*; an optimisation problem has overlapping subproblems when its main problem can be split up into a series of subproblems that can be reused multiple times (Demaine, 2011). A simple example of this is the problem of the Fibonacci numbers (see Explanation 2.2). Calculating the n^{th} Fibonacci number $F(n)$ requires knowledge of the values $F(n-1)$ and $F(n-2)$. These values, in turn, require $F(n-2)$, $F(n-3)$ and $F(n-4)$ to be calculated. $F(n-2)$ has however been calculated before and hence this subproblem is already solved;
- *Optimal substructure*; an optimisation problem has an optimal substructure when the optimal solution of the overall problem can be constructed from optimal solutions of its subproblems (Demaine, 2011). The easiest way to explain this is by considering both a shortest path and a longest path problem using, for example, the fictional network that is presented in Figure 2.5. For the shortest path problem the optimal substructure property holds, but this property does not hold for the longest path problem.

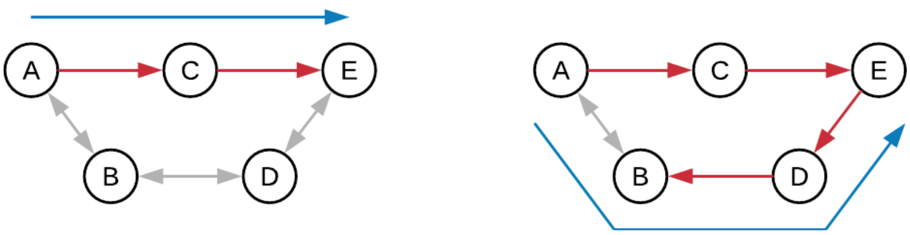


Figure 2.5: A fictional network for a shortest and longest path problem. The blue arrow in the left figure represents the shortest path from A to E, which clearly is a combination of the shortest paths from A to C and C to E (marked by the red arrows). The blue arrow in the right figure represents the longest path from A to E, which is not a combination of the longest paths from A to B, B to D and D to E, since the longest path from e.g. A to B runs via C, E and D (marked by the red arrows).

Explanation 2.2: Fibonacci Numbers

Fibonacci was an Italian mathematician (c. 1170 - c. 1250) that proposed a mathematical model to approximate the population growth of rabbits (Ghose, 2018). Although the model by Fibonacci did not become famous for this purpose, the sequence of numbers the model generates (see below) and especially the ratio between two successive values became very popular throughout the world. The so called golden ratio that is represented by ϕ and equals approximately 1.618 is found to be aesthetically beautiful and can frequently be found in nature too, such as in human genome DNA and in the arrangement of parts of leaves and branches along the stems of plants (Ghose, 2018).

$$F(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n \geq 2 \end{cases}$$

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}	F_{20}
0	1	1	2	3	5	8	13	21	34	55	89	144	233	377	610	987	1597	2584	4181	6765

In principle both of the dynamic programming properties are valid for traffic signal control. For example, if the delay is to be calculated for a set of control sequences of which only the final control decision between intervals $N-1$ and N differs, the delay until $N-1$ has to be calculated only once and can then be shared for all of the control sequences (overlapping subproblems). Also, the delay that a given optimal control sequence over N intervals causes is no different than the delay that the optimal control sequence over $N-1$ intervals causes if the delay of the optimal control sequence between the intervals of $N-1$ and N is added to that (optimal substructure). As a result, dynamic programming can be applied in traffic signal control.

In applying dynamic programming, the decision tree that includes all control options is typically searched in either one of the following two ways:

- In *backward recursion* or depth-first search the search algorithm starts with an initial control sequence at, e.g., depth d after which backtracking is performed to explore solutions that include the same higher level nodes (i.e. until $d-1$), but different nodes at the lower level d. If all options for d are explored, the procedure, which is sometimes also referred to as tabulation (Demaine, 2011), is repeated for higher level nodes (e.g. $d-2$ and $d-3$). Porche and Lafortune (1997) present a look-ahead control traffic signal control algorithm that relies on backward recursion;
- In *forward recursion* the search algorithm starts at the root node and iteratively explores nodes at increasing depths. Forward recursion requires a backward recovery procedure at the end of the optimisation to identify the optimal solution. Sen and Head (1997) present an example of forward recursion in a look-ahead traffic signal control algorithm. The algorithms by van Katwijk (2008) and Chen and Sun (2016) are very much based on the algorithm by Sen and Head (1997) and they also rely on forward recursion. All three algorithms basically split up the full horizon into smaller parts (subproblems) for which optimal solutions are calculated, and the overall optimal solution is then found by combining the optimal solutions of the subproblems.

2.3. Prediction Models

Mirchandani and Head (2001) and van Katwijk (2008) mention that a look-ahead traffic signal control algorithm requires models that are able to (a) predict vehicle arrivals and (b) evaluate the effects of control decisions in terms of delay and queueing. The new algorithm will be dependent on these models too and hence they are discussed in this section.

Vehicle Arrivals

In this study vehicle arrivals are obtained via the Run and Repeat approach that is explained in Section 1.5. Obviously, in practice vehicle arrivals cannot be obtained this way and hence the algorithm is dependent on either upstream loop detectors and linear extrapolation or car-following logic, or floating car data. Since upstream loop detectors are not always available and since Fan et al. (2010) and Goodall (2013) estimate that it will take at least 25 to 30 years before a high penetration rate of connected vehicles is achieved, the new algorithm should be able to function well without the availability of vehicle arrivals too. To prevent that too few vehicles are considered when determining the optimal sequence of control decisions (which could lead to green being terminated too early), one could think of adding vehicle arrivals manually on the basis of the estimated traffic flows. The Golden Controller already applies a similar method in the Golden Green module to determine maximum extension green times.

Queueing

With regard to queueing, two types of models are applied in traffic signal control. In *vertical* queueing models vehicles that cannot pass a certain bottleneck (e.g. a traffic signal) are stacked vertically and do not occupy any space (Knoop, 2017). In *horizontal* queueing models vehicles do occupy space and they thereby better represent actual queues. That is, horizontal queueing models take into account (a) the number of vehicles that can be stored on a road segment and (b) the queueing dissipation process in which the head of a queue behaves like a moving bottleneck while the tail of the queue may continue to grow (van Katwijk, 2008). Shock wave analysis as proposed by Knoop (2017) is often used in horizontal queueing.

Most of the existing algorithms use vertical queueing models in combination with static queue discharge rates that denote the rate at which vehicles leave a queue. The Golden Controller also applies a vertical queueing model. Queueing models of the vertical type are the easiest of the two and they require few computational effort. Anderson et al. (2014) argue that the outflow of traffic, which could be considered the most important output of queueing models for traffic signal control, is accurately predicted by both type of models. That is, the authors compared horizontal and vertical queueing models to observed trajectory data and found only slight differences in outflow estimation. Considering that the number of vehicles that fit on a lane, which may be defined as a second relevant aspect of queueing models for traffic signal control, could be incorporated in vertical queueing by the introduction of a new variable, it seems logical to apply vertical queueing in the new traffic signal control algorithm too.

Delay

Most of the existing look-ahead traffic signal control algorithms use relatively simple mathematical formulations to evaluate the consequences of control decisions. Goodall (2013) considers a different approach and applies microsimulation to assess control decisions. Although microsimulation might be slightly more accurate, it also goes hand in hand with a high computational load and as a result microsimulation does not easily allow for expansion into longer time horizons. Although Feng et al. (2015) argue that the algorithm by Goodall (2013) is already unable to run in real time due to the computational requirements of parallel simulation, increasing the planning horizon would certainly not improve this (Goodall (2013) only considers a horizon of fifteen to twenty seconds). It seems obvious to apply simple mathematical formulations in the new algorithm too.

2.4. Summary

The findings enumerated below summarise the chapter:

- Traffic signal control algorithms rely on a rolling horizon approach for optimisation over time. The number of intervals of the horizon and the interval size are important design choices and their values are to be chosen wisely. A horizon that is too short might lead to short-sighted control decisions and non-optimality, while a horizon that is too long unnecessarily increases computational load. A resolution that is too small leads - in combination with a long horizon - to a very large decision space which in turn results in high computational load, while a resolution that is too large decreases the level of detail of the planning. To speed up the optimisation process a variable resolution could be introduced in which the interval size increases as the planning progresses into the future. Also, by carefully choosing when the algorithm recalculates, unnecessary recalculations can be prevented, thereby achieving a lower update frequency which implies less strict computational requirements;

- Search algorithms, which represent the way in which the decision space is searched through, form another important design choice. Dynamic programming, which is a constructive search method that breaks up a problem into a number of subproblems, allows for reasoning about whole classes of similar assignments and it is thereby able to reduce the search space, which in turn decreases computation time. For this reason dynamic programming is often found in look-ahead traffic signal control and it seems obvious to make the new algorithm rely on this too;
- Most of the existing algorithms that have been reviewed in this chapter cannot (easily) be reused within the new traffic signal control algorithm. They namely either present complicated network-oriented approaches or they have certain inefficiencies incorporated. The algorithms by Sen and Head (1997), van Katwijk (2008) and Chen and Sun (2016), which are all dynamic programming formulations, consider single intersections and they efficiently explore the decision space for the sequence of control decisions that best fits a control objective. They also present relatively clear approaches. As a result of this, these algorithms are candidates to be reused in the new algorithm. Section 3.1 elaborates on the exact implementation of these algorithms within the new algorithm;
- Like any look-ahead traffic signal control algorithm, the new algorithm is dependent on models to predict vehicle arrivals and to evaluate control decisions. In practice vehicle arrivals are not always available and they might have to be added manually on the basis of e.g. traffic flows. In approximating queueing the application of a vertical queueing model and static vehicle discharge rates seems obvious. Such kind of models are relatively simple and they accurately approximate queueing. It makes sense to use simple mathematical formulations to evaluate control decisions in terms of delay, as other approaches such as microsimulation do not easily meet computational requirements;
- Hardly any of the authors of the algorithms reviewed for this study and referred to in this chapter mention anything about time-to-green or time-to-red predictions. The only authors that do mention this topic consider it future work. It has to be mentioned, however, that when a planning of control decisions is known, time-to-green and time-to-red are only derivatives;
- None of the scientific works that have been mentioned in this chapter elaborate on applying a planning of control decisions in practice and adjusting these control decisions on the basis of actual traffic conditions.

3

Planning of Control Decisions Algorithm

This chapter is dedicated to the description of the Planning of Control Decisions (PCD) algorithm. This algorithm plans a sequence of control decisions on the basis of current queues and expected vehicle arrivals. Section 3.1 starts the chapter by discussing the algorithm's set-up. This section forms the link with the literature review of Chapter 2 and describes how the PCD algorithm relates to existing look-ahead traffic signal control algorithms. In Section 3.2 vehicle arrivals are defined and corresponding principles are discussed. Section 3.3 describes the algorithm's search technique for finding feasible control policies and Section 3.4 defines mathematical formulations for the evaluation of control decisions. Section 3.5 describes how the control policy that best fits a control objective can be obtained using a backtracking procedure and this section also describes the actual output that is generated by the PCD algorithm. Section 3.6 explains how time-to-green and time-to-red predictions are obtained for individual signal groups and Section 3.7 elaborates on the algorithm's computation time. Finally, Section 3.8 summarises the chapter's main findings.

3.1. Algorithm Set-up

Of all of the algorithms that have been evaluated in Chapter 2, the algorithms by Sen and Head (1997), van Katwijk (2008) and Chen and Sun (2016), referred to as [SH], [vK] and [CS] respectively, are most useful for this study. Although these algorithms share the principle of splitting up the planning horizon into smaller parts and combining solutions for these smaller parts to find the overall optimal solution, they differ in various ways too. The PCD algorithm includes the best parts of these algorithms and it adds a number of new functions. The next bullet points clarify how the new algorithm precisely relates to the above mentioned algorithms.

- The PCD algorithm is a dynamic programming formulation that adopts the basic forward recursion optimisation technique that is introduced by [SH] and is also applied by [vk] and [CS]. This technique prevents that the full decision space has to be explored, which saves a lot of computation time. Depending on the precise implementation, this search technique even allows for finding the overall optimal solution. The optimality of the PCD algorithm is discussed in Section 3.3.3;
- Like [SH] and [CS] the PCD algorithm considers stages instead of blocks[1] and it applies a signal group-based approach that is somewhat similar to [vK] for the evaluation of control decisions. By doing so, the planning matches the flexibility that standard Dutch actuated traffic signal control offers with primary ahead realisations, which is beneficial for both delay reduction and time-to-green/time-to-red prediction accuracy, while it remains possible to explore a large range of control options. An elaborate explanation of the differences between a stage-based and a signal group-based approach can be found later in this section. The introduction of stages in a signal-group based approach requires mathematical formulations that take into account that signal groups can be present in two successive stages, which means that the formulations that [vK] defines throughout his work cannot simply be copied;
- Unlike [SH] and [CS] which consider only the number of vehicles in a queue and unlike [vK] which records estimated arrival and departure times of vehicles, the PCD algorithm keeps track of the composition and weight of a queue. This means that any form of priority can be given on the level of signal groups, vehicle classes and even individual vehicles. Among other things the following can then be achieved:

[1] A signal group can be present in multiple stages, while a signal group is only included once in a set of blocks.

  - Certain vehicle classes can be prioritised (e.g. emergency vehicles and public transport, but also heavy loaded lorries if they can be detected);
  - Priority can be given to vehicles of which the waiting time has exceeded a predefined value. This can also be done on the level of signal groups;
  - It can be prevented that signal groups are given green for too many times in a row before other signal groups are given green. This is especially relevant for major-minor type of intersections.
- Like [vK] the PCD algorithm includes signal group-dependent formulations for queue dissolvement on the basis of a vertical queueing model and user-adjustable queue discharge rates. [SH] and [CS] also apply vertical queueing, but they do not present any formulations for queue dissolvement;
- Additional to [SH], [vK] and [CS] the PCD algorithm incorporates the possible condition at the start of recalculation where no signal group is currently given green. The PCD algorithm also allows for not granting the right on green to the signal groups of any stage if there is no (predicted) traffic demand at the intersection. An example of this is provided in Figure 3.1. By keeping the planning open signal groups that experience unexpected traffic demand can be given green, which reduces the vulnerability of the PCD algorithm for incorrect vehicle arrivals;

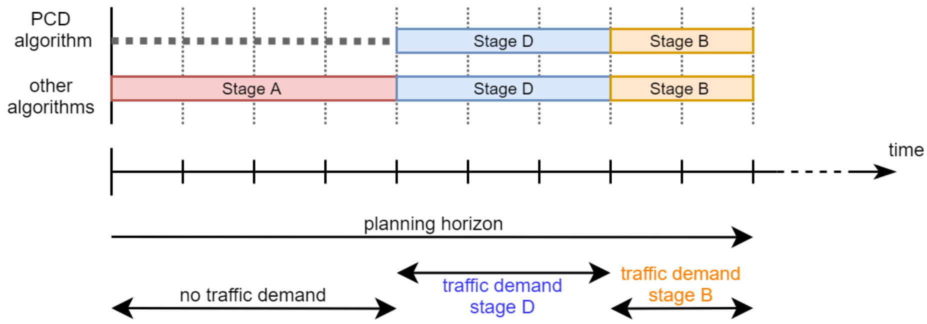


Figure 3.1: The PCD algorithm keeps the planning of control decisions open (gray colour) in case there are no queues and no expected vehicle arrivals. Other algorithms do not include this possibility and they typically dedicate the leftover time to the first stage of the set of stages.

- Additional to [SH], [vK] and [CS] the PCD algorithm produces the expected delay and queues at the end of the planning horizon, which allows for comparison of the planning with other plannings;
- Additional to [SH], [vK] and [CS] the PCD algorithm produces green start and green end times for all of the considered signal groups and provides road users with time-to-green and time-to-red predictions accordingly, including a certainty indication.

Stage-based versus Signal group-based Approach
The main difference between the approaches by Sen and Head (1997), van Katwijk (2008) and Chen and Sun (2016) relates to how individual signal groups are treated within the algorithm. To clarify this, let's consider the example intersection that is presented in Figure 3.2 together with its corresponding set of blocks and stages.

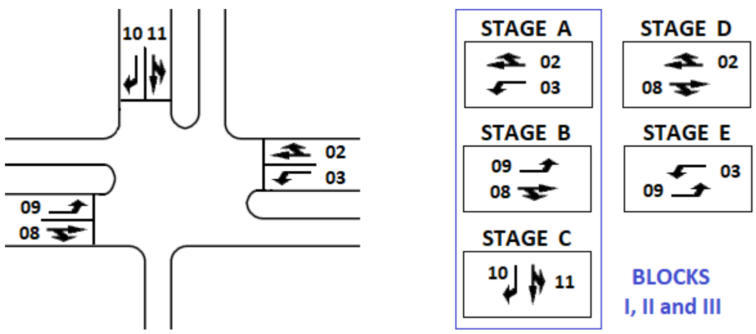


Figure 3.2: An example intersection with corresponding sets of blocks and stages.

In a stage-based approach, such as the ones by Sen and Head (1997) and Chen and Sun (2016), it is assumed that any signal group in a stage may not be green parallel to a signal group in any other stage. A stage-based approach also assumes that all signal groups in a stage are given green at the start and until the end of a stage. In this type of approach stages are considered as single units and they are basically just put after each other in time. An intergreen interval, which is typically the same value for any two stages, is included in between the stages. Figure 3.3 visualises a stage-based approach. If a signal group is in two successive stages, green can be lengthened cosmetically over the intergreen interval as Figure 3.4 shows.

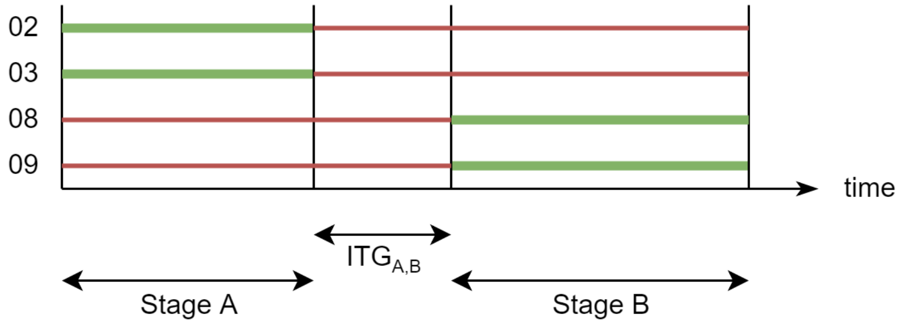


Figure 3.3: An example of a planning of green intervals for signal groups according to a stage-based approach. $ITG_{A,B}$ denotes the intergreen interval between stages A and B. The green colour denotes green time and the red colour denotes red time. Amber is included in the red time.

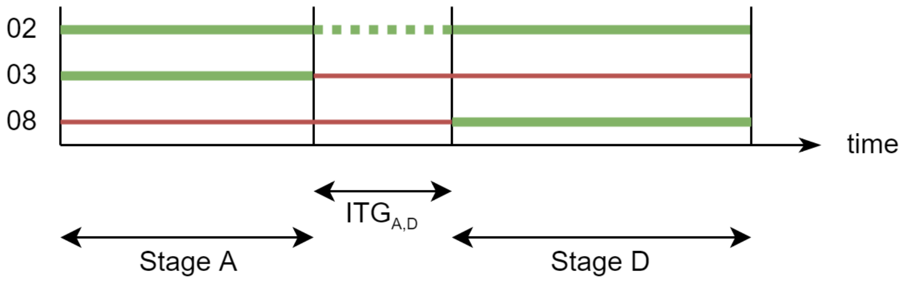


Figure 3.4: An example of a planning of green intervals for signal groups according to a stage-based approach. $ITG_{A,D}$ denotes the intergreen time between stages A and D. The green time of signal group 02 is cosmetically lengthened over the intergreen interval between stages A and D.

The benefit of a stage-based approach is that it is relatively simple. However, some serious disadvantages come into play when this type of approach is applied to Dutch traffic signal control. That is, due to signal groups realising ahead of time, deviations arise between the theoretical planning and the execution of the planning. These deviations especially occur when there is a large variety in intergreen times between the individual signal groups of a stage. In the latter case, defining only one intergreen time that is valid for the entire stage is very inconvenient. An example of this can be observed in Figure 3.5. The problem of a stage-based approach becomes even more apparent if green of a signal group in a stage can be ended earlier than green of another signal group in that same stage (see Figure 3.6).

In the more complicated signal group-based approach that van Katwijk (2008) proposes, it is assumed that a signal group of a stage can be given green parallel to a signal group of another stage. This situation may occur if green of a signal group can be ended (there is no more traffic), while green of another signal group in the same stage cannot yet be ended (traffic is still present), which allows a signal group that is located in the next stage to realise ahead of time. The benefit of such an approach is that the planning more accurately represents the execution of the planning via standard Dutch actuated traffic signal control, which is beneficial for time-to-green and time-to-red prediction accuracy. Also, there is no need to somehow translate intergreen times between signal groups into intergreen times on an inconvenient level of stages.

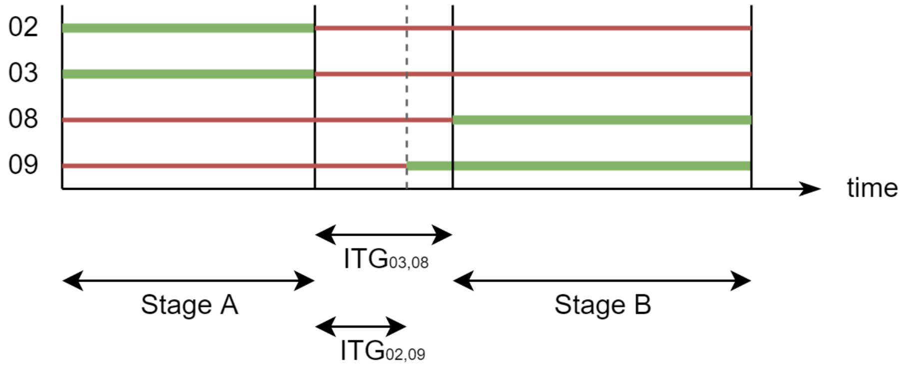


Figure 3.5: The stage-based planning of Figure 3.3 executed via Dutch actuated traffic signal control. The different intergreen times between signal groups allows signal group 09 to become green earlier than signal group 08 and as a result deviations arise between the theoretical planning and the execution of the planning.

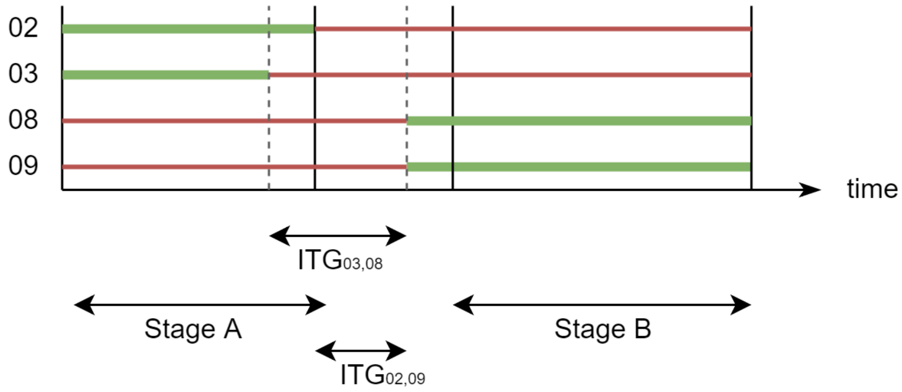


Figure 3.6: The stage-based planning of Figure 3.3 executed via Dutch actuated traffic signal control. The different intergreen times and the early green end of signal group 03 allow signal groups 08 and 09 to realise earlier, and as a result deviations arise between the theoretical planning and the execution of the planning.

Decision Space Reduction via a Signal group-based Approach
Using a signal group-based approach in which signal groups of different stages can be given green simultaneously, van Katwijk (2008) claims that the computation time can be reduced, while the quality of the solution is kept at a constant level. The author namely argues that blocks can be applied over stages[2], thereby decreasing the number of control options that need to be checked. In case of the example intersection of Figure 3.2, only three blocks would be required to represent all five stages. Although it is true that all five stages can be created out of the three blocks (thanks to planning signal groups ahead of time), in practice the application of blocks limits the set of solutions that can be achieved. The next example clarifies why this is the case.

Let's consider Figure 3.2 again and let's assume that the signal groups of stage C/block III (10 and 11) are currently given green. Furthermore, let's assume that there is a green request for signal groups 02, 03, 08 and 09 and let's assume that from the point of view of delay reduction it is best to give green to signal groups 02 and 08 first right after green of signal groups 10 and 11 finishes. Although signal groups 02 and 08 can indeed be realised together using blocks I and II, namely after either signal group 03 or 09 has become green, they will not be realised together right after green ending of signal groups 10 and 11, because they do not form a block together. Including an additional block, which in essence means the application of stages rather than blocks, is the only way to check for this control option too in a signal-group based approach.

[2] Although the definition of a stage is the same as the definition of a block, namely 'a set of signal groups that can be given green at the same time', there is an important difference between the two. This difference relates to how many times signal groups are included in a set of blocks/stages. Signal groups are typically included in multiple stages, while signal groups are only present once in a set of blocks. This effectively means that the number of stages is larger than the number of blocks.

3.2. Vehicle Arrivals and Queueing

In this section vehicle arrivals are defined and their relationship with queueing is explained. The variables that are denoted in Table 3.1 are introduced for this. Please note that the variables in this table are also introduced throughout the text and hence Table 3.1 is for reference purposes only.

Variable	*Domain*	*Unit*	*Description*
$A_{\lambda,k}$	$\mathbb{N}_0$	-	The number of vehicles that are expected to arrive at signal group λ at time k
k	$\mathbb{N}_1$	s	Time index
λ	$\mathbb{N}_1$	-	Signal group index
t	$\mathbb{R}$	s	Real time
T	$\mathbb{R}$	s	Model time step size

Table 3.1: A list of variables, their domains, their units and their descriptions. The interval size that is considered in this study equals one second. In theory, though, the PCD algorithm is able to handle any positive interval size.

One of the key aspects of the new traffic signal control algorithm is that it incorporates predictions of vehicle arrivals. Apart from increasing the predictability of control decisions, the idea behind this is that the new algorithm further reduces the total delay (and number of stops) of road users by taking into account vehicles that are close to the intersection, but that have not yet been spotted by the loop detectors (these are the so called vehicle arrivals). By doing so, the new algorithm is thought to better anticipate onto arriving vehicles than traditional traffic signal control algorithms are able to.

To achieve a reduction of the delay, the new algorithm is required to know when vehicles really need a green signal to pass the intersection. In essence this means that a vehicle arrival should represent that one moment in time at which a vehicle reaches the stop line. In this study Definition 3.1 is hence applied for a vehicle arrival.

Definition 3.1: Vehicle Arrival

A signal group λ receives a vehicle arrival at time t if and only if there is a vehicle that is expected to arrive at the stop line of signal group λ at time t.

To determine when a vehicle arrives at the stop line, geographic/locational data of that vehicle is required. In this study this data consists of the distance between the current location of the vehicle and the location of the stop line of the signal group the vehicle approaches, combined with the vehicle's velocity. In this study[3] the Run and Repeat approach that is explained in Section 1.5 is applied to obtain locational data.

In implementing vehicle arrivals into the PCD algorithm, the principles that are enumerated below are considered along with the definition of time that is presented in Figure 3.7. In this figure time index k is the model time and it represents the time period $t \in [k \cdot T, (k+1) \cdot T)$, where $T \in \mathbb{R}$ is the model time step size in seconds. In this study time intervals of one second are considered, hence $T = 1$.

- A vehicle that arrives at the stop line of signal group λ within the time interval $[t, t+1)$ is set to arrive at this signal group at time $k+1$. As an example, the vehicle that arrives at the stop line of signal group 02 in Figure 3.8 at $t \approx 7.3$ is considered a vehicle arrival for signal group 02 at time $k = 8$ (hence $A_{02,8} = 1$);
- A vehicle that arrives at signal group λ at time t joins the already existing queue at signal group λ if its corresponding signal is not equal to green at time t. The vehicle that arrives at signal group 08 at $t \approx 5.3$ (thus at $k = 6$, see Figure 3.8 again), joins the queue at signal group 08 before it is able to discharge from the signal group's queue when its signal turns green at time $k = 7$ (hence in the time interval $t = [6, 7)$);
- A vehicle that arrives at signal group λ at time t is immediately able to discharge from a queue according to the signal group's queue discharge rate if its corresponding signal is green at time t. For example, the vehicle that arrives at signal group 03 at $t \approx 2.7$ (hence at time $k = 3$) is able to discharge from the queue right away since the signal of signal group 03 in Figure 3.8 is green at $k = 3$.

[3] As mentioned in Section 1.5, there are other approaches next to Run and Repeat that could be thought of to obtain vehicle arrivals.

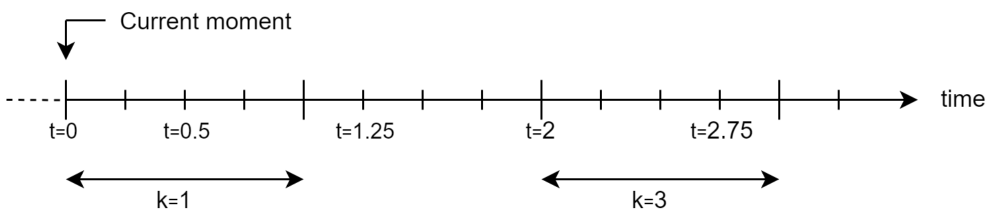


Figure 3.7: The definition of time within the PCD algorithm. t Represents the real time, which is part of the set of real numbers ($t \in \mathbb{R}$), and k represents the time index, which is part of the set of natural numbers ($k \in \mathbb{N}_1$).

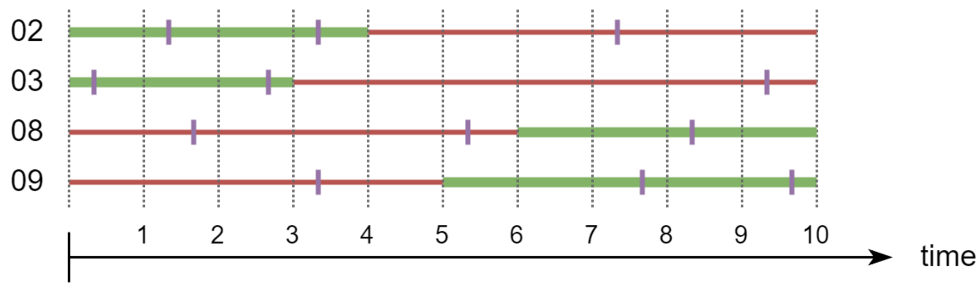


Figure 3.8: The green lines represent intervals of green time. The purple dashes represent arriving vehicles.

Table 3.2 represents the vehicle arrivals as presented in Figure 3.8. Table 3.3 presents the signal statuses for the control policy that is visualised in Figure 3.8.

Time	*02*	*03*	*08*	*09*
1	0	1	0	0
2	1	0	1	0
3	0	1	0	0
4	1	0	0	1
5	0	0	0	0
6	0	0	1	0
7	0	0	0	0
8	1	0	0	1
9	0	0	1	0
10	0	1	0	1

Table 3.2: Aggregated vehicle arrivals for signal groups 02, 03, 08 and 09 corresponding to the arriving vehicles in Figure 3.8. Vehicle arrivals are always integers.

Time	*02*	*03*	*08*	*09*
1	G	G	-	-
2	G	G	-	-
3	G	G	-	-
4	G	-	-	-
5	-	-	-	-
6	-	-	-	G
7	-	-	G	G
8	-	-	G	G
9	-	-	G	G
10	-	-	G	G

Table 3.3: Aggregated signal statuses corresponding to the control policy of Figure 3.8. 'G' denotes green time and '-' denotes amber/red time.

3.3. Search Algorithm

In this section the dynamic programming algorithm is described that searches through the decision space for control policies. The variables that the algorithm relies on are introduced throughout the paragraphs of this section and for reference purposes they can also be found in Tables 3.4 and 3.5 and in Appendix D.

The goal of the algorithm is to find the sequence and duration of stages for which the best performance is achieved over the full planning horizon h (main problem). Since there are many candidate solutions, the decision space is in fact exposed to exponential growth, and since there are strict computational requirements, simply looping through the full set of control options cannot be achieved within the limited amount of time that is available. Finding the best control policy is thus no straightforward task. The algorithm that is presented in this section tries to simplify the main problem by breaking it down into a number of smaller, more manageable subproblems and it does this by considering the interrelated concepts of *iterations* and *limited horizons*.

Variable	*Domain*	*Unit*	*Description*
$c(s,x,j,\lambda,k)$	-	-	The queue composition of signal group λ in iteration j at time interval k given a limited horizon s and a control decision x. Each of the entries in this vector represents the type of vehicle. Example: $[Car, Car, Bus, Car]$
$C^*_{s,j,\lambda}$	-	-	The queue composition of signal group λ that corresponds to the best control decision for limited horizon s in iteration j. Each of the entries in this vector represents the type of vehicle. Example: $[Car, Car, Bus, Car]$
$d(s,x,j,\lambda,k)$	$\mathbb{N}_0$	s	The delay of signal group λ in iteration j for time interval k given a limited horizon s and a control decision x
$D^*_{s,j,\lambda}$	$\mathbb{N}_0$	s	The delay for signal group λ that corresponds to the best control decision for limited horizon s in iteration j
FC	-	-	A user-defined ordered set of signal groups.
$ge(s,x,j,\lambda)$	$\mathbb{Z}$	s	The green end time of signal group λ in iteration j given a limited horizon s and control decision x
$gs(s,x,j,\lambda)$	$\mathbb{Z}$	s	The green start time of signal group λ in iteration j given a limited horizon s and control decision x
$GE^*_{s,j,\lambda}$	$\mathbb{Z}$	s	The green end time of signal group λ that corresponds to the best control decision for limited horizon s in iteration j
$GS^*_{s,j,\lambda}$	$\mathbb{Z}$	s	The green start time of signal group λ that corresponds to the best control decision for limited horizon s in iteration j
h	$\mathbb{N}_1$	s	Full planning horizon length
j	$\mathbb{N}_1$	-	Iteration counter of the PCD algorithm
$j^{\min}$	$\mathbb{N}_1$	-	The absolute minimum number of iterations that the PCD algorithm has to perform to find the best control policy
k	$\mathbb{N}_1$	s	Time index
λ	$\mathbb{N}_1$	-	Signal group index
N^{stages}	$\mathbb{N}_1$	-	The number of stages in the ordered set SG. This variable hence denotes the cardinality of SG
φ	$\mathbb{N}_1$	-	Stage index
$q(s,x,j,\lambda,k)$	-	-	The queue fraction of signal group λ in iteration j at time interval k given a limited horizon s and a control decision x. Each of the entries in this vector represents the fraction of a vehicle that is still present in the queue. Example: $[0.4,1,1,1]$
$Q^*_{s,j,\lambda}$	-	-	The queue fraction of signal group λ that corresponds to the best control decision for limited horizon s in iteration j. Each of the entries in this vector represents the fraction of a vehicle that is still present in the queue. Example: $[0.4,1,1,1]$
s	$\mathbb{N}_1$	s	Limited horizon
SG_φ	-	-	A user-defined ordered set of stages. Stage φ denotes a selection of signal groups $\lambda \in FC$
t^{fg}_λ	$\mathbb{N}_1$	s	The fixed green time of signal group λ. The fixed green time represents the minimum green time
$w(s,x,j,\lambda,k)$	-	-	The queue weight of signal group λ in iteration j at time interval k given a limited horizon s and a control decision x. Each of the entries in this vector represents the weight of a vehicle. Example: $[1,1,3,1]$
$W^*_{s,j,\lambda}$	-	-	The queue weight of signal group λ that corresponds to the best control decision for limited horizon s in iteration j. Each of the entries in this vector represents the weight of a vehicle. Example: $[1,1,3,1]$

Table 3.4: A list of variables, their domains, their units and their descriptions.

Variable	*Domain*	*Unit*	*Description*
x	$\mathbb{N}_0$	s	Control option; x denotes the number of time intervals that are allocated to a stage
$X^*_{s,j}$	$\mathbb{N}_0$	s	The number of time intervals that are allocated to a stage in the best control policy for limited horizon s in iteration j
$X(s)$	-	-	An ordered set of feasible control options x given a limited horizon s
z	$\mathbb{N}_0$	s	The number of time intervals that is left to allocate to the stage or combination of stages that form the best control policy for the corresponding limited horizon in the previous iteration $j-1$

Table 3.5: A list of variables, their domains, their units and their descriptions.

Iterations
Instead of taking into account the full set of stages SG[4] all at once, the PCD algorithm considers stages, denoted φ, individually in iterations. In the first iteration $j = 1$ only the first stage is considered, in the second iteration $j = 2$ only the second stage is considered, and this process continues onward. During each iteration j the algorithm evaluates the performance effect of allocating green time to the signal groups of stage $\varphi = j \bmod N^{\text{stages}}$. The algorithm determines itself when it has found the best control policy and there is no need to specify the number of iterations that need to be executed beforehand. The algorithm cycles over stages, hence starts over again at stage $\varphi = 1$, in case the optimal solution has not been found when the final stage of the ordered set SG is considered in iteration $j = N^{\text{stages}}$.

Limited horizons
During each iteration j, the algorithm splits up the full horizon h into smaller parts that could effectively be seen as limited horizons s (see Figure 3.9). The algorithm starts at the smallest limited horizon $s = 1$ and iteratively increases s until the full horizon h is reached. The algorithm thereby considers all limited horizons $s \in \{1, \dots, h\}$.

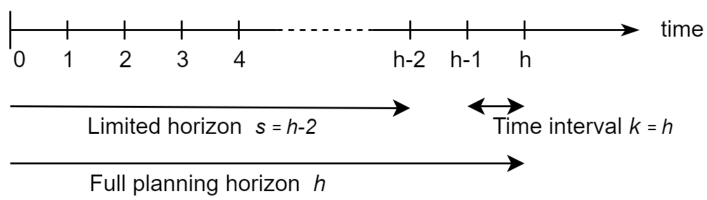


Figure 3.9: The full planning horizon h is split up into limited horizons s.

For each limited horizon $s \in \{1, \dots, h\}$ the algorithm determines a set of feasible control options $X(s)$ via:

$$X(s) = \begin{cases} \{0, g^{\min}, \dots, s\} & \text{if } s \geq g^{\min} \\ \{0\} & \text{otherwise} \end{cases} \tag{3.1}$$
$$\text{where:} \quad g^{\min} = \min(\{\, t^{\text{fg}}_{\lambda} : \lambda \in SG_{j \bmod N^{\text{stages}}} \}\,)$$

The ordered set $X(s)$ basically ranges from $x = 0$, where the signal groups of stage $\varphi = j \bmod N^{\text{stages}}$ are skipped, to $x = s$, where all of the available time within the limited horizon s is allocated to the signal groups of stage $\varphi = j \bmod N^{\text{stages}}$. By only allowing control options to be evaluated for which the minimum green time constraint is satisfied for at least one signal group $\lambda \in SG_{j \bmod N^{\text{stages}}}$, computation time is reduced[5].

[4] SG is a self-defined set of stages, where each stage consists of a set of signal groups that can be given green at the same time. SG may include all possible combinations of signal groups, but this is certainly not required. It holds that the greater the number of stages, the larger the number of control options that are considered and hence the better the solution may become. At the same time, the larger the number of stages, the larger the computation time becomes (see Section 3.7).

[5] Signal groups in $SG_{j \bmod N^{\text{stages}}}$ for which the minimum green time constraint cannot be met within control decision x will simply not be allowed to become green, as will be explained in Section 3.4.

Although determining control options via Equation 3.1 seems pretty uncomplicated, it allows for incorporating one of the most important features in adaptive traffic signal control. That is, by always evaluating a control decision $x = 0$ the performance effect of entirely skipping a stage is evaluated, even though the signal groups of this stage may request green. This means that although the algorithm cycles through stages in a fixed order (i.e. $A \to B \to C \to D \to A \to \dots$ in case SG consists of four stages), any order of stages may eventually end up to be the best performing control policy (e.g. $C \to A \to D$).

Each control option $x \in X(s)$ gives a coarse indication of how long the signal groups of stage $\varphi = j \bmod N^{\text{stages}}$ may be given green. The actual green time is, however, different for each signal group $\lambda \in SG_{j \bmod N^{\text{stages}}}$. This green time namely depends on (a) the green end time of conflicting signal groups λ^* and (b) the intergreen time between these conflicting signal groups λ^* and signal group λ. By providing four examples, Figures 3.11, 3.12, 3.13 and 3.14 clarify how control decision x should be interpreted.

The key to the algorithm is that when x is insufficient to fill the limited horizon s, then there is time to fill in before x starts. This leftover time is denoted z and allows the signal groups of another stage or combination of stages to become green. The leftover time z is calculated via:

$$z = s - x \tag{3.2}$$

By allocating the leftover time z to the stage or combination of stages that was found to be the best performing solution for limited horizon z in the previous iteration $j - 1$, the solution continuously improves. Figure 3.10 indicates how the variables s, x and z relate to each other.

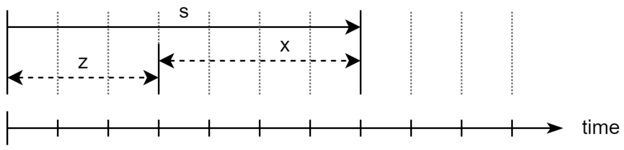


Figure 3.10: The relationship between the variables s, x and z.

After evaluating the performance effect of each control decision $x \in X(s)$ via the equations in Section 3.4, the control option for which the best performance is achieved is stored as $X^*_{s,j}$. This control decision x either equals 0 or any integer $g^{\min} \leq x \leq h$. The corresponding performance for each signal group $\lambda \in FC$ is stored as $D^*_{s,j,\lambda}$. This information can then be reused in next iterations of the algorithm.

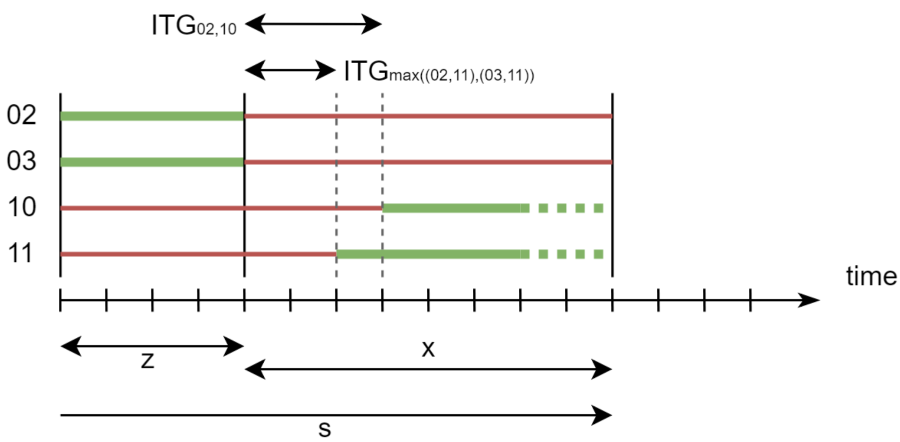


Figure 3.11: Control decision x gives a coarse indication of how long the signal groups of stage $\varphi = j \bmod N^{\text{stages}}$ may be given green. The actual green time for each of the signal groups in this stage may however be different. This green time namely depends on the green end time of conflicting signal groups and the intergreen time. The starting point is that green of a signal group $\lambda \in SG_{j \bmod N^{\text{stages}}}$ is realised the earliest moment possible. This means that, in the case visualised above, green of signal groups 10 and 11 starts respectively 3 and 2 time intervals later than the start of x. The broken green line indicates that green may end earlier than the end time of x (which is s), as long as the minimum green time constraint is met. This is elaborated on in Section 3.4. Green can never end later than s.

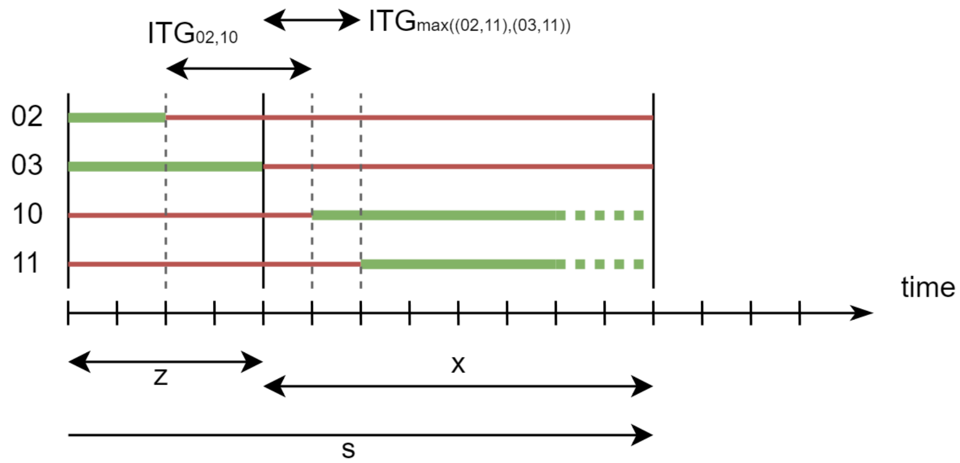


Figure 3.12: A premature ending of green of signal group 02 allows signal group 10 to realise two time intervals earlier. See the caption of Figure 3.11 for further details.

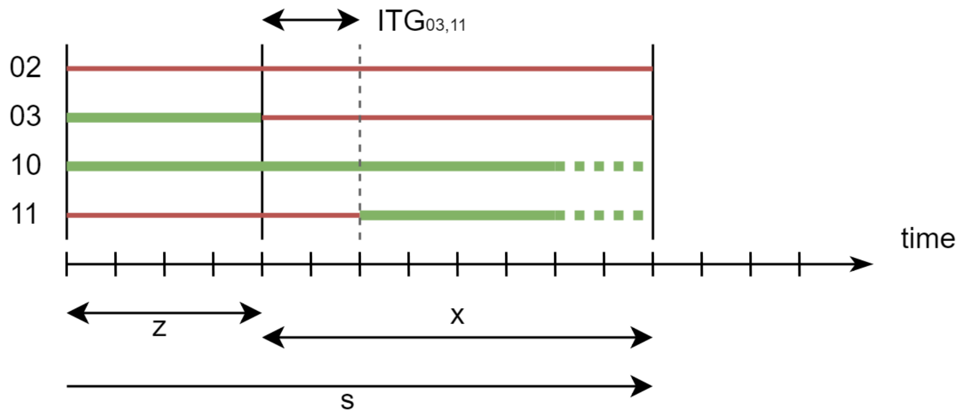


Figure 3.13: The absence of a green request for signal group 02 allows signal group 10 to become green earlier than the start of x. See the caption of Figure 3.11 for further details.

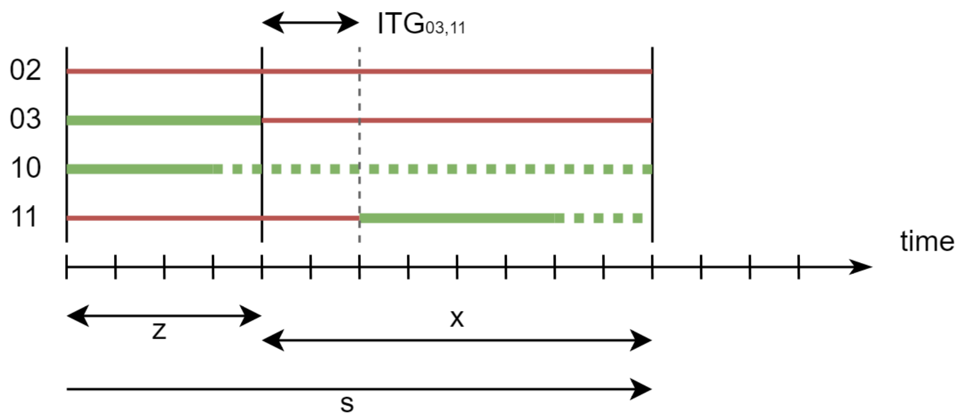


Figure 3.14: The absence of a green request for signal group 02 allows signal group 10 to become green earlier than the start of x. Green of signal group 10 may even end earlier than the start of x, as long as the minimum green time constraint for this signal group is met. See the caption of Figure 3.11 for further details.

3.3.1. Stop Criteria

As mentioned earlier, specification of the number of iterations over which optimisation has to take place is not required since the algorithm itself determines when it is finished. The algorithm has found the best control policy if the stage that is considered in the next iteration is the same stage that lastly introduced a better performance for any limited horizon $s \in \{1,\dots,h\}$, while the number of iterations that have already been performed is at least equal to the number of stages in SG (N^{stages}). This means that the algorithm continues iterating over stages in iterations j as long as at least one of the following two criteria is met:

- $j \leq N^{\text{stages}}$
- $D^*_{s,j-N^{\text{stages}}} \neq D^*_{s,j} \quad \forall s \in \{1,\dots,h\}$

The actual and absolute maximum number of iterations that the PCD algorithm needs to perform to find the best control policy is a function of the number of stages in SG, the vehicle arrivals and the number of times signal groups can be given green within the full horizon h and cannot easily be calculated. On the contrary, the absolute minimum number of iterations only depends on the number of stages in SG and is calculated via:

$$j^{\text{min}} = N^{\text{stages}} \tag{3.3}$$

3.3.2. Complete Search Algorithm

Algorithm 3.1 describes the full search algorithm as a summary of Section 3.3.

Algorithm 3.1: Complete search algorithm of the PCD algorithm

Input : FC, h, N^{stages}, SG
Output : Sequence of control decisions

set iteration $j = 1$
while $j \leq N^{\text{stages}} \vee D^*_{s,j-N^{\text{stages}},\lambda} \neq D^*_{s,j,\lambda} \forall s \in \{1,\dots,h\}$ **do**
  foreach $s \in \{1,\dots,h\}$ **do**
    determine the set of control options $X(s)$ via Equation 3.1
    foreach $x \in X(s)$ **do**
      determine z via Equation 3.2
      foreach $\lambda \in FC$ **do**
        determine the delay $d(s,x,j,\lambda)$, queues $c(s,x,j,\lambda)$, $q(s,x,j,\lambda)$ and $w(s,x,j,\lambda)$, green starts $gs(s,x,j,\lambda)$ and green ends $ge(s,x,j,\lambda)$ via the equations in Section 3.4
      end
    end
    determine the policy that leads to the lowest cumulative delay over all signal groups $\lambda \in FC$, store the corresponding delay per signal group as $D^*_{s,j,\lambda}$ and store the corresponding control decision as $X^*_{s,j}$
    for each $\lambda \in FC$ store the queues that correspond to the control decision with the lowest overall delay as $C^*_{s,j,\lambda}$, $Q^*_{s,j,\lambda}$ and $W^*_{s,j,\lambda}$ and store the corresponding green starts and green ends as $GS^*_{s,j,\lambda}$ and $GE^*_{s,j,\lambda}$
  end
  update iteration $j \leftarrow j+1$
end
apply the backtracking procedure that is described in Section 3.5 on $X^*_{s,j}$ to obtain the optimal sequence of control decisions

3.3.3. Optimality of the Solution

Due to how the PCD algorithm is defined, it cannot be guaranteed that the overall optimal control policy is found. As explained in the next paragraphs, this has to do with (a) the signal group-based approach that is considered and (b) the way in which intergreen times are included.

In the Dutch approach to traffic signal control intergreen times are considered on the level of individual signal groups. This, together with the possibility for signal groups to realise ahead of time, allows for efficient traffic signal control. In the optimisation process of the PCD algorithm the intergreen time between signal groups is incorporated in the x-part (see Figures 3.11, 3.12, 3.13 and 3.14). This is a requirement, since it is unknown beforehand which stage will succeed the final stage of the z-part, and hence it is unknown what intergreen times to include. To guarantee finding the overall optimal solution, intergreen times should however have been included in the z-part.

The algorithms by Sen and Head (1997) and Chen and Sun (2016) do find the overall optimal solution, however, this is only possible because they consider (a) a stage-based approach in which signal groups cannot be given green ahead of time and (b) a single intergreen time that is valid between any two signal groups, which allows for taking the intergreen time into account within the z-part. As explained in Section 3.1, both of these principles are inconvenient for Dutch actuated/adaptive traffic signal control since they introduce inaccuracies and inefficiencies.

3.4. Performance Functions

To evaluate control decisions, performance functions are required. In this section these performance functions are described. The generic notation that is denoted in Tables 3.6, 3.7 and 3.8 is introduced.

Variable	*Domain*	*Unit*	*Description*
$A_{\lambda,k}$	$\mathbb{N}_0$	-	The number of vehicles that are expected to arrive at signal group λ at time k
$A_{\lambda,k}^{\text{composition}}$	-	-	The composition of the vehicle that are expected to arrive at signal group λ at time interval k. Each of the entries in this vector represents the type of vehicle. Example: $[Car, Car, Bus, Car]$. By definition the number of elements in this vector equals $A_{\lambda,k}$
$A_{\lambda,k}^{\text{fraction}}$	-	-	The fraction of the vehicles that are expected to arrive at signal group λ at time interval k. Each of the entries in this vector represents the fraction of a vehicle that is still present. By definition each entry in this vector equals 1 and the number of elements equals $A_{\lambda,k}$. Example: $[1,1,1,1]$
$A_{\lambda,k}^{\text{weight}}$	-	-	The weight factor of the vehicles that are expected to arrive at signal group λ at time interval k. Each of the entries in this vector represents the weight of a vehicle. Example: $[1,1,3,1]$. By definition the number of elements in this vector equals $A_{\lambda,k}$
$c_{\lambda}^{\text{initial}}$	-	-	The queue composition at the start of recalculation of the PCD algorithm. Each of the entries in this vector represents the type of vehicle. Example: $[Car, Car, Bus, Car]$
$c(s,x,j,\lambda,k)$	-	-	The queue composition of signal group λ in iteration j at time interval k given a limited horizon s and a control decision x. Each of the entries in this vector represents the type of vehicle. Example: $[Car, Car, Bus, Car]$
$C_{s,j,\lambda}^{*}$	-	-	The queue composition of signal group λ that corresponds to the best control decision for limited horizon s in iteration j. Each of the entries in this vector represents the type of vehicle. Example: $[Car, Car, Bus, Car]$
$d(s,x,j,\lambda,k)$	$\mathbb{N}_0$	s	The delay of signal group λ in iteration j for time interval k given a limited horizon s and a control decision x

Table 3.6: A list of variables, their domains, their units and their descriptions.

Variable	*Domain*	*Unit*	*Description*
$D^*_{s,j,\lambda}$	$\mathbb{N}_0$	s	The delay for signal group λ that corresponds to the best control decision for limited horizon s in iteration j
FC	-	-	A user-defined ordered set of signal groups
$ge(s,x,j,\lambda)$	$\mathbb{Z}$	s	The green end time of signal group λ in iteration j given a limited horizon s and control decision x
$ge^{\text{initial}}_{\lambda}$	$\mathbb{Z}$	s	The green end time of signal group λ at the start of recalculation of the PCD algorithm
$GE^*_{s,j,\lambda}$	$\mathbb{Z}$	s	The green end time of signal group λ that corresponds to the best control decision for limited horizon s in iteration j
$gs(s,x,j,\lambda)$	$\mathbb{Z}$	s	The green start time of signal group λ in iteration j given a limited horizon s and control decision x
$GS^*_{s,j,\lambda}$	$\mathbb{Z}$	s	The green start time of signal group λ that corresponds to the best control decision for limited horizon s in iteration j
h	$\mathbb{N}_1$	s	Full planning horizon length
j	$\mathbb{N}_1$	-	Iteration counter of the PCD algorithm
k	$\mathbb{N}_1$	s	Time index
λ	$\mathbb{N}_1$	-	Signal group index
μ_λ	$\mathbb{N}_1$	veh/h	The queue discharge rate for signal group λ
N^{stages}	$\mathbb{N}_1$	-	The number of stages in the ordered set SG. This variable hence denotes the cardinality of SG
φ	$\mathbb{N}_1$	-	Stage index
$q^{\text{initial}}_{\lambda}$	-	-	The queue fraction of signal group λ at the start of recalculation of the PCD algorithm. Each of the entries in this vector represents the fraction of a vehicle that is still present in the queue. Example: $[0.4,1,1,1]$
$q(s,x,j,\lambda,k)$	-	-	The queue fraction of signal group λ in iteration j at time interval k given a limited horizon s and a control decision x. Each of the entries in this vector represents the fraction of a vehicle that is still present in the queue. Example: $[0.4,1,1,1]$
$Q^*_{s,j,\lambda}$	-	-	The queue fraction of signal group λ that corresponds to the best control decision for limited horizon s in iteration j. Each of the entries in this vector represents the fraction of a vehicle that is still present in the queue. Example: $[0.4,1,1,1]$
s	$\mathbb{N}_1$	s	Limited horizon
SG_φ	-	-	A user-defined ordered set of stages. Stage φ denotes a selection of signal groups λ of set FC
$t^{\text{cl}}_{\lambda,\lambda^*}$	$\mathbb{Z}$	s	The clearance time between signal groups λ and λ^*. A clearance time of -99 denotes that there is no conflict between signal groups λ and λ^*
t^{fg}_{λ}	$\mathbb{N}_1$	s	The fixed green time of signal group λ. The fixed green time represents the minimum green time
t^{ge}_{λ}	$\mathbb{N}_0$	s	The latest moment at which green of signal group λ must end
t^{gs}_{λ}	$\mathbb{N}_0$	s	The earliest moment at which green of signal group λ can start
t^{rg}_{λ}	$\mathbb{N}_1$	s	The guaranteed red time of signal group λ
t^{y}_{λ}	$\mathbb{N}_1$	s	The set amber time of signal group λ
$w^{\text{initial}}_{\lambda}$	-	-	The queue weight of signal group λ at the start of recalculation of the PCD algorithm. Each of the entries in this vector represents the weight of a vehicle. Example: $[1,1,3,1]$

Table 3.7: A list of variables, their domains, their units and their descriptions.

Variable	*Domain*	*Unit*	*Description*
$w(s,x,j,\lambda,k)$	-	-	The queue weight of signal group λ in iteration j at time interval k given a limited horizon s and a control decision x. Each of the entries in this vector represents the weight of a vehicle. Example: $[1,1,3,1]$
$W^*_{s,j,\lambda}$	-	-	The queue weight of signal group λ that corresponds to the best control decision for limited horizon s in iteration j. Each of the entries in this vector represents the weight of a vehicle. Example: $[1,1,3,1]$
wf_λ	$\mathbb{R}$	-	The weight factor for signal group λ
x	$\mathbb{N}_0$	s	Control option; x denotes the number of time intervals that are allocated to a stage
$X(s)$	-	-	An ordered set of feasible control options x given a limited horizon s
$X^*_{s,j}$	$\mathbb{N}_0$	s	The number of time intervals that are allocated to a stage in the best control policy for limited horizon s in iteration j

Table 3.8: A list of variables, their domains, their units and their descriptions.

In evaluating control decisions, one can distinguish four cases. These cases are enumerated below and the importance of distinguishing them has to do with the availability of data from previous iterations on the one hand (case 1 versus cases 2, 3 and 4) and the lack for the need of complicated equations for determining the performance of a control decision on the other hand (cases 2 and 3 versus case 4).

1. Iteration $j = 1$;
2. Iteration $j > 1$, $x = 0$;
3. Iteration $j > 1$, $x = s$;
4. Iteration $j > 1$, others.

3.4.1. Case 1 ($j = 1$)

The total delay of any signal group λ is defined as the sum of the queue fraction q times the queue weight w over all time intervals k that are considered in the limited horizon s. In mathematical terms this translates to:

$$d(s,x,j,\lambda) = \sum_{k=1}^{s} q(s,x,j,\lambda,k)^{\mathrm{T}} \cdot w(s,x,j,\lambda,k) \tag{3.4}$$

When determining the delay d for all of the individual signal groups $\lambda \in FC$, one has to consider two different situations:

- Either signal group λ is not part of the stage that is granted the right on green in the current iteration j, hence $\lambda \notin SG_{j \bmod N^{\text{stages}}}$, which means that the signal(s) of signal group λ will stay red for all of the time intervals $k \in \{1,\ldots,s\}$. Any queue that exists at the current moment will remain present and will increase in size if vehicles are expected to arrive;
- Or signal group λ is part of the stage that is granted the right on green in the current iteration j, hence $\lambda \in SG_{j \bmod N^{\text{stages}}}$, which means that the signals of λ may turn green if the minimum green time constraint is met within control decision x.

In the first situation, where the signal(s) of signal group λ remain red, the queues c, q and w are calculated via:

$$\begin{aligned} c(s,x,j,\lambda) &= [c^{\text{initial}}_\lambda, [A^{\text{composition}}_{\lambda,1}, \ldots, A^{\text{composition}}_{\lambda,s}]] \\ q(s,x,j,\lambda) &= [q^{\text{initial}}_\lambda, [A^{\text{fraction}}_{\lambda,1}, \ldots, A^{\text{fraction}}_{\lambda,s}]] \\ w(s,x,j,\lambda) &= [w^{\text{initial}}_\lambda, [(A^{\text{weight}}_{\lambda,1}, \ldots, A^{\text{weight}}_{\lambda,s}]] \cdot wf_\lambda \end{aligned} \tag{3.5}$$

These equations basically state that all of the vehicles to are expected to arrive within the limited horizon s are added to the initial queues. Since green is not given, the green start and end times do not have to be updated, which means that the following applies:

$$
\begin{aligned}
gs(s,x,j,\lambda) &= gs_{\lambda}^{\text{initial}} \\
ge(s,x,j,\lambda) &= ge_{\lambda}^{\text{initial}}
\end{aligned}
\tag{3.6}
$$

A visual representation of this first situation is given by [1] in Figure 3.15.

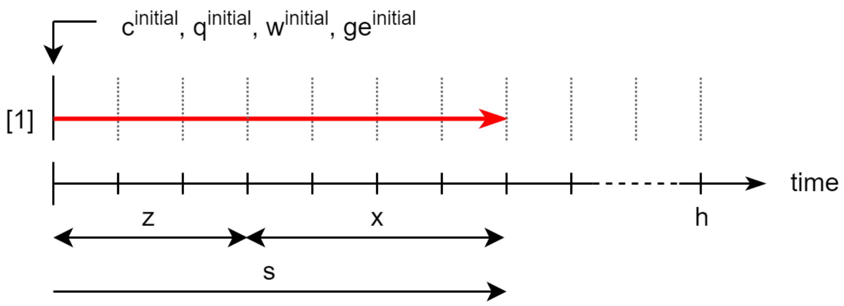


Figure 3.15: Visual representation of the time intervals $k \in \{1,\ldots,s\}$ over which the delay d is calculated for any signal group λ that may not become green.

In the second situation signal group λ does have the right on green and as a result its signal may become green if there is sufficient time within the control decision x to at least realise the fixed green time t_{λ}^{fg}. To find out if there is sufficient time for a green realisation, the following needs to be determined:

- The earliest moment t at which the signal(s) of signal group λ may turn green. This is denoted t_{λ}^{gs} and is calculated via Algorithm 3.2. This algorithm basically loops over all signal groups $\lambda^* \in FC$ and if a signal group λ^* conflicts with signal group λ ($t_{\lambda^*,\lambda}^{\text{cl}} > -99$), then it considers the green end time of signal group λ^* and it adds to that the amber time $t_{\lambda^*}^{\text{y}}$ and the clearance time $t_{\lambda^*,\lambda}^{\text{cl}}$. If $\lambda = \lambda^*$ and λ is currently not green, then the amber time (t_{λ}^{y}) and guaranteed red time (t_{λ}^{rg}) are incorporated.

Algorithm 3.2: The algorithm for determining the earliest time at which the signal(s) of signal group λ may turn green.

Input : FC, $ge_{\lambda}^{\text{initial}}$, j, s, t_{λ}^{y}, t_{λ}^{rg}, $t_{\lambda^*,\lambda}^{\text{cl}}$, x
Output: t_{λ}^{gs}

set $t_{\lambda}^{\text{gs}} = s - x$ if $j = 1$, set $t_{\lambda}^{\text{gs}} = 0$ otherwise
foreach $\lambda^* \in FC$ **do**
    if $\lambda^* = \lambda$ **then**
        if $ge_{\lambda}^{\text{initial}} < 0$ **then**
            $t_{\lambda}^{\text{gs}} = \max(t_{\lambda}^{\text{gs}}, ge_{\lambda}^{\text{initial}} + t_{\lambda}^{\text{y}} + t_{\lambda}^{\text{rg}})$
        end
    else
        if $t_{\lambda^*,\lambda}^{\text{cl}} > -99$ **then**
            $t_{\lambda}^{\text{gs}} = \max(t_{\lambda}^{\text{gs}}, ge_{\lambda}^{\text{initial}} + t_{\lambda^*}^{\text{y}} + t_{\lambda^*,\lambda}^{\text{cl}})$
        end
    end
end

Please note that in the first iteration $j = 1$ green cannot start earlier than the start of x (see line 1 of Algorithm 3.2). This is necessary to incorporate the possibility to not give green to the signal groups of any stage if there are no current queues ($q_\lambda^{\text{initial}} = \emptyset$) and no expected vehicle arrivals ($\max(A_{\lambda,1},\ldots,A_{\lambda,s}) = 0$). For iterations $j > 1$ green may start the earliest moment possible ($t_\lambda^{\text{gs}} = 0$);

- The latest moment t at which the signal(s) from signal group λ must turn red. This is denoted t_λ^{ge} and is naturally equal to s, hence $t_\lambda^{\text{ge}} = s$.

If the amount of time between the earliest moment green may start and the last moment green must end is smaller than the fixed green time, thus $t_\lambda^{\text{ge}} - t_\lambda^{\text{gs}} < t_\lambda^{\text{fg}}$, then there is insufficient time for a green realisation of signal group λ. In this situation the queues c, q and w and green start and end times gs and ge are determined according to the procedure that explained earlier this section, hence via Equations 3.5 and 3.6. If there is sufficient time to realise green, then the queues c, q and w and the green start and end time gs and ge are determined via Algorithm 3.3. This algorithm shows how queues evolve over time as a function of the queue in the previous time step $k-1$, the arrivals at time k and queue discharging at time k. Like for the other signal groups, the delay for signal group λ can be determined via Equation 3.4 using the output of Algorithm 3.3.

Algorithm 3.3: The algorithm for determining queues and the green end time if the signal(s) of a signal group λ are allowed to become green.

Input : $A_{\lambda,k}$, $A_{\lambda,k}^{\text{composition}}$, $A_{\lambda,k}^{\text{fraction}}$, $A_{\lambda,k}^{\text{weight}}$, $c_\lambda^{\text{initial}}$, $\text{gs}(s,x,j,\lambda)$, μ_λ, $q_\lambda^{\text{initial}}$, s, t_λ^{fg}, t_λ^{gs}, $w_\lambda^{\text{initial}}$, wf_λ
Output: $c(s,x,j,\lambda)$, $\text{ge}(s,x,j,\lambda)$, $q(s,x,j,\lambda)$, $w(s,x,j,\lambda)$

set $c(s,x,j,\lambda,0) = c_\lambda^{\text{initial}}$, $q(s,x,j,\lambda,0) = q_\lambda^{\text{initial}}$, $w(s,x,j,\lambda,0) = w_\lambda^{\text{initial}}$ and $gs(s,x,j,\lambda) = t_\lambda^{\text{gs}}$
foreach $k \in \{1,\ldots,s\}$ **do**
    if $A_{\lambda,k} > 0$ **then**
        $[c(s,x,j,\lambda,k), A_{\lambda,k}^{\text{composition}}]$
        $[q(s,x,j,\lambda,k), A_{\lambda,k}^{\text{fraction}}]$
        $[w(s,x,j,\lambda,k), (A_{\lambda,k}^{\text{weight}} * wf_\lambda)]$
    end
    if $k \in \{t_\lambda^{\text{gs}}+1,\ldots,s\}$ **then**
        if $c(s,x,j,\lambda,k) \neq \emptyset$ **then**
            $c(s,x,j,\lambda,k) = \max(c(s,x,j,\lambda,k-1) \ominus \mu_\lambda, \emptyset)$
            $q(s,x,j,\lambda,k) = \max(q(s,x,j,\lambda,k-1) \ominus \mu_\lambda, \emptyset)$
            $w(s,x,j,\lambda,k) = \max(w(s,x,j,\lambda,k-1) \ominus \mu_\lambda, \emptyset)$
            $ge(s,x,j,\lambda) = k$
        else
            if $ge(s,x,j,\lambda) - gs(s,x,j,\lambda) < t_\lambda^{\text{fg}}$ **then**
                $ge(s,x,j,\lambda) = k$
            end
        end
    end
end

As can be observed in Algorithm 3.3, calculation of the queues is split up into two parts. Between time intervals $k = 1$ and $k = t_\lambda^{\text{gs}}$ the signal(s) of signal group λ are red (no vehicles may discharge) and between intervals $k = t_\lambda^{\text{gs}} + 1$ and $k = s$ the signal(s) are green, and hence vehicles do have the possibility to discharge from the queue. [2] And [3] in Figure 3.16 present visual examples of this.

The green end time of signal group λ is defined to be the moment at which there is no queue left, while the minimum green time constraint is met, while considering that green can never end later than s. This means that green of signal group λ does not necessarily have to be extended until s, but instead may end earlier too as is presented by [4] in Figure 3.16. This opens up the possibility to realise signal groups of the then unknown successive stage ahead of time.

Please note that the ⊖ sign in Algorithm 3.3 means that vehicles are removed from the queues c, q and w according to the queue discharge rate of signal group λ (μ_λ). Further details on how vehicles are precisely removed from the queues can be found in the algorithm's code.

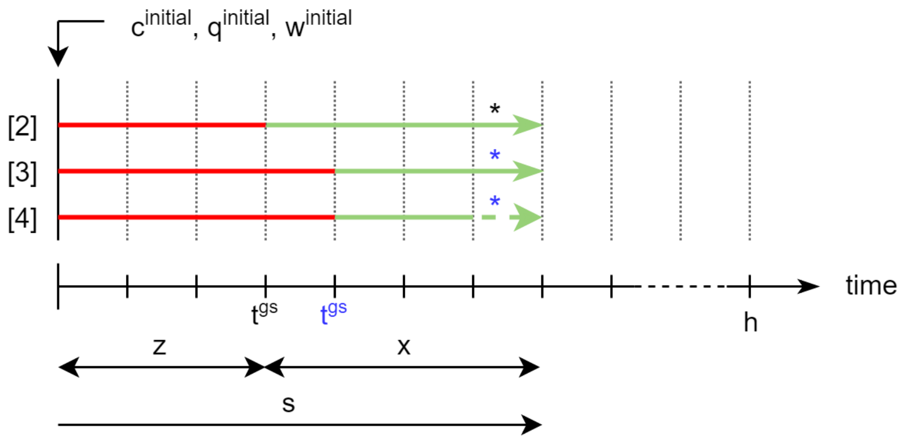


Figure 3.16: Visual representation of the time intervals k over which the delay is calculated for any signal group λ that may become green within the limited horizon s.

After control decision x has been evaluated for all individual signal groups $\lambda \in FC$, the total delay of that control decision can be calculated by summing the delay over all signal groups $\lambda \in FC$. For this purpose the following equation is applied:

$$d(s,x,j) = \sum_{\lambda \in FC} d(s,x,j,\lambda) \tag{3.7}$$

3.4.2. Case 2 ($j > 1$, $x = 0$)

If the control option is evaluated for which no time is allocated to stage $\varphi = j \bmod N^{\text{stages}}$, then the delay, queues, green start time and green end time can simply be obtained from the best solution of the previous iteration $j-1$. This means that the following equations apply:

$$d(s,x,j) = \sum_{\lambda \in FC} D^*_{s,j-1,\lambda} \qquad x = 0, \quad \forall j > 1 \tag{3.8}$$

$$\begin{aligned} c(s,x,j,\lambda) &= C^*_{s,j-1,\lambda} && x = 0, \quad j > 1, \quad \forall \lambda \in FC \\ q(s,x,j,\lambda) &= Q^*_{s,j-1,\lambda} && x = 0, \quad j > 1, \quad \forall \lambda \in FC \\ w(s,x,j,\lambda) &= W^*_{s,j-1,\lambda} && x = 0, \quad j > 1, \quad \forall \lambda \in FC \end{aligned} \tag{3.9}$$

$$\begin{aligned} gs(s,x,j,\lambda) &= GS^*_{s,j-1,\lambda} && x = 0, \quad j > 1, \quad \forall \lambda \in FC \\ ge(s,x,j,\lambda) &= GE^*_{s,j-1,\lambda} && x = 0, \quad j > 1, \quad \forall \lambda \in FC \end{aligned} \tag{3.10}$$

A visual representation of the second case is given by [5] in Figure 3.17.

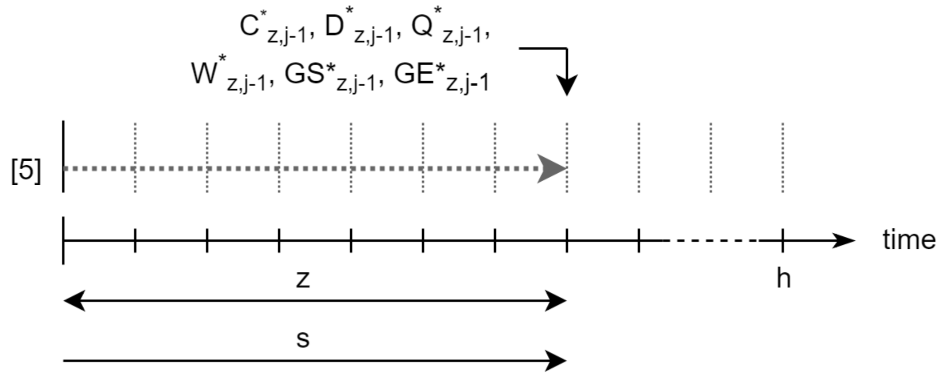


Figure 3.17: In the situation that $x = 0$, thus $s = z$, no time is allocated to the signal groups of stage $\varphi = j \bmod N^{\text{stages}}$, and hence no calculations need to take place. In this situation the delay, queues, green starts time and green ends time equal the solution for limited horizon z from the previous iteration $j-1$.

3.4.3. Case 3 ($j > 1$, $x = s$)

In the case that the maximum amount of time is allocated to stage $\varphi = j \bmod N^{\text{stages}}$, then the delay d, queues c, q and w, the green start times gs and the green end times ge can be calculated according to the same procedure that holds for the first iteration $j = 1$. This means that:

- The delay, queues and green starts/ends of any signal group $\lambda \in FC$ that may not or cannot be given green can be calculated via Equations 3.4, 3.5 and 3.6;
- The delay, queues and green starts/ends of any signal group $\lambda \in FC$ that may and can be given green can be calculated via Equation 3.4 and Algorithm 3.3;
- The total delay of the control decision x can be calculated via Equation 3.7.

3.4.4. Case 4 ($j > 1$, $x =$ other)

In the fourth and final case procedures are largely the same as the ones that have been explained for case 1, however, there are differences in terms of the time intervals over which the delay is calculated. These differences originate from the principle that information from the previous iteration $j-1$ is incorporated, as explained in Section 3.3. The interval differences are first explained for signal groups which may not turn green, and they are explained for signal groups that do have the possibility to become green thereafter. In any case, though, it holds that the delay for a signal group $\lambda \in FC$ can be calculated using:

$$d(s,x,j,\lambda) = D^*_{z,j-1,\lambda} + \sum_{k=z+1}^{s} q(s,x,j,\lambda,k)^{\mathrm{T}} \cdot w(s,x,j,\lambda,k) \tag{3.11}$$

In the situation that the delay is calculated for a signal group λ that may not become green, the queues are determined via:

$$\begin{aligned} c(s,x,j,\lambda) &= [C^*_{z,j-1,\lambda},[A^{\text{composition}}_{\lambda,z+1},\ldots,A^{\text{composition}}_{\lambda,s}]] \\ q(s,x,j,\lambda) &= [Q^*_{z,j-1,\lambda},[A^{\text{fraction}}_{\lambda,z+1},\ldots,A^{\text{fraction}}_{\lambda,s}]] \\ w(s,x,j,\lambda) &= [W^*_{z,j-1,\lambda},[A^{\text{weight}}_{\lambda,z+1},\ldots,A^{\text{weight}}_{\lambda,s}]] \cdot wf_\lambda \end{aligned} \tag{3.12}$$

This equation basically states that the queues that belong to the best performing solution for limited horizon z in the previous iteration $j-1$ are picked, and the vehicles that are expected to arrive between intervals $k = z+1$ and $k = s$ are simply added. A visual representation of this is provided by [6] in Figure 3.18. The green start and end times are also obtained from the previous iteration $j-1$ via:

$$\begin{aligned} gs(s,x,j,\lambda) &= GS^*_{s,j-1,\lambda} \\ ge(s,x,j,\lambda) &= GE^*_{s,j-1,\lambda} \end{aligned} \tag{3.13}$$

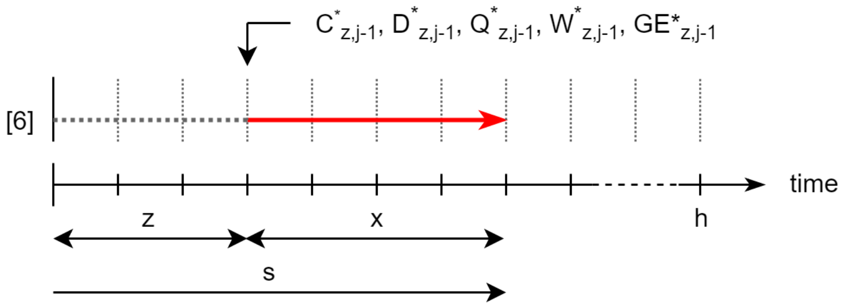


Figure 3.18: The delay is calculated for all time intervals k that are included in x (red arrow), taking the queues at the start of x as a basis. The delay over these intervals is then added to the delay that corresponds to the best performing control policy for limited horizon z in iteration $j-1$ (represented by the broken gray line). This gives the total delay over all time intervals $k \in \{1,\ldots,s\}$.

In the situation that a signal group λ may become green, then it is again calculated at what particular moment in time the signal(s) of signal group λ may turn green. For this purpose Algorithm 3.2 is reused.

In case the green start time of signal group λ is equal to or later than the start time of x, then the delay and queues are calculated in the standard way using Algorithm 3.3 for the intervals between $k = z+1$ and $k = s$. Here, the starting conditions for the calculations are formed by $C^*_{z,j-1,\lambda}$, $D^*_{z,j-1,\lambda}$, $GE^*_{z,j-1,\lambda}$, $GS^*_{z,j-1,\lambda}$, $Q^*_{z,j-1,\lambda}$ and $W^*_{z,j-1,\lambda}$. A visualisation of this is provided by [8] and [9] in Figure 3.19.

In the case that signal group λ may realise ahead of time, i.e. green of signal group λ starts earlier than the start of x, then the delay $D^*_{z,j-1,\lambda}$ needs to be compensated for. That is, in calculating the delay for the best performing control policy of limited horizon z in iteration $j-1$ it was unjustly assumed that vehicles may not discharge and hence too much delay was incorporated into $D^*_{z,j-1,\lambda}$.

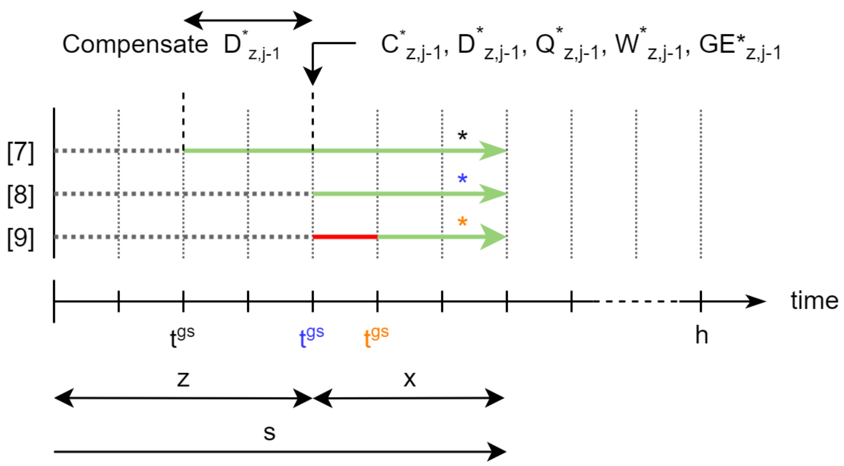


Figure 3.19: The delay is calculated for all time intervals k that are included in x (green and green/red arrows), taking the queues at the start of x as a basis. The delay over these intervals is then added to the delay that corresponds to the best performing control policy for limited horizon z in iteration $j-1$ (represented by the broken gray line). This gives the total delay over all time intervals $k \in \{1,\ldots,s\}$. If green of a signal group λ starts earlier than the start of x, delay compensation for that signal group takes place, since it was unjustly assumed while calculating $D^*_{z,j-1,\lambda}$ that vehicles may not discharge from the queue.

Compensating for wrongly adjusted delay works as follows. First, the intervals are determined over which delay compensation is to take place. These intervals equal the difference between t^{gs}_{λ} and the start of x. Then, with the compensation intervals being known, the queues at the first compensation interval are recreated. This is done by subtracting the number of expected vehicle arrivals over all compensation intervals from the queues at the start of x. The queues at the start of x are namely known and they are equal to $C^*_{z,j-1,\lambda}$, $Q^*_{z,j-1,\lambda}$ and $W^*_{z,j-1,\lambda}$. Then, using the queues at the start first compensation interval as a basis, the delay is calculated over all compensation intervals. This delay is then subtracted from $D^*_{z,j-1,\lambda}$, after which calculation of the new delay and new queues takes place over all time intervals between the green start time (t^{gs}_{λ}) and the green end time (s) again using the queues at the first interval of compensation as a starting point. For precise mathematical formulations, please consider Algorithm 3.4. Please note that the $\ominus$ sign in this algorithm means that a number of vehicles are subtracted from queues C, Q and W that corresponds to $A_{\lambda,t^{\text{gs}}_{\lambda}\to z}$.

Algorithm 3.4: The algorithm for determining the amount of delay with which $D^*_{z,j-1,\lambda}$ needs to be compensated.

Input : $A_{\lambda,k}$, t^{gs}_{λ}, z
Output: $D^*_{s,j-1,\lambda}$

if $t^{\text{gs}}_{\lambda} < z$ **then**
  determine the number of vehicles that are expected to arrive over all compensation intervals via $A_{\lambda,t^{\text{gs}}_{\lambda}\to z} = \sum_{k=t^{\text{gs}}_{\lambda}+1}^{z} A_{\lambda,k}$
  obtain the queues at the first time interval over which compensation is to take place via $C_{s,j-1,\lambda} = C_{s,j-1,\lambda} \ominus A_{\lambda,t^{\text{gs}}_{\lambda}\to z}$, $Q_{s,j-1,\lambda} = Q_{s,j-1,\lambda} \ominus A_{\lambda,t^{\text{gs}}_{\lambda}\to z}$ and $W_{s,j-1,\lambda} = W_{s,j-1,\lambda} \ominus A_{\lambda,t^{\text{gs}}_{\lambda}\to z}$
  use $C_{s,j-1,\lambda}$, $Q_{s,j-1,\lambda}$ and $W_{s,j-1,\lambda}$ as input in Equation 3.3 to calculate the delay over time intervals $k \in \{t^{\text{gs}}_{\lambda},\ldots,z\}$ and denote this delay as $D_{\lambda,t^{\text{gs}}_{\lambda}\to z}$
  compensate the delay via $D_{s,j-1,\lambda} = D_{s,j-1,\lambda} - D_{\lambda,t^{\text{gs}}_{\lambda}\to z}$
end

3.4.5. Best Performing Control Decision

After having determined the performance for all signal groups $\lambda \in FC$, the part of the algorithm in which a single control decision x is evaluated is finished. The algorithm now continues evaluating the remaining control decisions $x \in X(s)$. After all $x \in X(s)$ have been evaluated, the algorithm picks and stores the best performing control decision x for limited horizon s by applying:

$$\begin{aligned}
D^*_{s,j,\lambda} &= d(s,x,j,\lambda) \text{ for which } \sum_{\lambda\in FC} d(s,x,j,\lambda) = \min\Big(\sum_{\lambda\in FC} d(s,x_1,j,\lambda),\ldots,\sum_{\lambda\in FC} d(s,x_n,j,\lambda)\Big)\\
X^*_{s,j} &= x \text{ for which } \sum_{\lambda\in FC} d(s,x,j,\lambda) = \min\Big(\sum_{\lambda\in FC} d(s,x_1,j,\lambda),\ldots,\sum_{\lambda\in FC} d(s,x_n,j,\lambda)\Big)
\end{aligned} \tag{3.14}$$

The equations above apply for all $s \in \{1,\ldots,h\}$ and for any iteration j. The corresponding queues c, q and w, green start times gs and green end times ge are determined and stored via:

$$\begin{aligned}
C^*_{s,j,\lambda} &= c(s,x,j,\lambda) \text{ for which } \sum_{\lambda\in FC} d(s,x,j,\lambda) = \min\Big(\sum_{\lambda\in FC} d(s,x_1,j,\lambda),\ldots,\sum_{\lambda\in FC} d(s,x_n,j,\lambda)\Big)\\
Q^*_{s,j,\lambda} &= c(s,x,j,\lambda) \text{ for which } \sum_{\lambda\in FC} d(s,x,j,\lambda) = \min\Big(\sum_{\lambda\in FC} d(s,x_1,j,\lambda),\ldots,\sum_{\lambda\in FC} d(s,x_n,j,\lambda)\Big)\\
W^*_{s,j,\lambda} &= c(s,x,j,\lambda) \text{ for which } \sum_{\lambda\in FC} d(s,x,j,\lambda) = \min\Big(\sum_{\lambda\in FC} d(s,x_1,j,\lambda),\ldots,\sum_{\lambda\in FC} d(s,x_n,j,\lambda)\Big)
\end{aligned} \tag{3.15}$$

$$\begin{aligned}
GS^*_{s,j,\lambda} &= gs(s,x,j,\lambda) \text{ for which } \sum_{\lambda\in FC} d(s,x,j,\lambda) = \min\Big(\sum_{\lambda\in FC} d(s,x_1,j,\lambda),\ldots,\sum_{\lambda\in FC} d(s,x_n,j,\lambda)\Big)\\
GE^*_{s,j,\lambda} &= ge(s,x,j,\lambda) \text{ for which } \sum_{\lambda\in FC} d(s,x,j,\lambda) = \min\Big(\sum_{\lambda\in FC} d(s,x_1,j,\lambda),\ldots,\sum_{\lambda\in FC} d(s,x_n,j,\lambda)\Big)
\end{aligned} \tag{3.16}$$

The equations basically state that x, c, q, w, ge and ge are picked for the control policy x for which the delay d over all signal groups $\lambda \in FC$ is minimum. If multiple policies have equal minimum delays, then the first policy the algorithm comes across is stored. This is the policy for which the x is lowest, which is convenient because it leaves more room for other signal groups to be given green.

3.5. Retrieval of the Control Policy

As soon as the stop criterion is reached, the control policy can be obtained that fits the control objective best using a backtracking procedure on $X^*_{s,j}$. The generic notation that is denoted in Table 3.9 and applies for the backtracking procedure.

Variable	*Domain*	*Unit*	*Description*
CP^{duration}	-	-	A vector containing the optimal duration of stages. Example: $[8,13,5]$
CP^{ge}_λ	-	-	A vector containing the green end times for signal group λ according to the optimal planning of control decisions. Example: $[9,35]$
CP^{gs}_λ	-	-	A vector containing the green start times for signal group λ according to the optimal planning of control decisions. Example: $[4,28]$
CP^{stages}	-	-	A vector containing the optimal sequence of stages. Example: $[B,A,C]$
FC	-	-	A user-defined ordered set of signal groups
h	$\mathbb{N}_1$	s	Full planning horizon length
h^{bt}	$\mathbb{N}_0$	s	Backtrack horizon. This value indicates how many time intervals from h remain to be included into the optimal control policy
j	$\mathbb{N}_1$	-	Iteration counter of the PCD algorithm
j^*	$\mathbb{N}_1$	-	The final iteration j that was performed in the PCD algorithm
λ	$\mathbb{N}_1$	-	Signal group index
N^{stages}	$\mathbb{N}_1$	-	The number of stages in the ordered set SG. This variable hence denotes the cardinality of SG
φ	$\mathbb{N}_1$	-	Stage index
s	$\mathbb{N}_1$	s	Limited horizon
SG_φ	-	-	A user-defined ordered set of stages. Stage φ denotes a selection of signal groups λ of set FC
$X^*_{s,j}$	$\mathbb{N}_0$	s	The number of time intervals that are allocated to a stage in the best control policy for limited horizon s in iteration j

Table 3.9: A list of variables, their domains, their units and their descriptions.

The backtracking algorithm works as follows. The algorithm loops through all of the iterations j in descending order and starts at $j = j^* - (N^{\text{stages}} - 1)$. In this iteration the control policy for the full horizon h has namely been updated lastly, which means that was found. At each iteration j the algorithm considers the control decision in $X^*_{h^{\text{bt}},j}$, where h^{bt} is set equal to the full horizon length h.

If the control decision $X^*_{h^{\text{bt}},j}$ equals zero, then the stage that was granted the right on green in iteration j is skipped and h^{bt} remains unchanged. The algorithm then simply proceeds to preceding iteration $j-1$.

If the control decision is non-zero ($X^*_{h^{\text{bt}},j} > 0$), then the stage that was granted the right on green in iteration j is part of the control policy and hence stage $\varphi = j \bmod N^{\text{stages}}$ is appended to CP^{stages}. The corresponding duration of the control policy is appended to CP^{duration} and the green start and end times for each signal group $\lambda \in FC$ are appended to CP^{gs}_λ and CP^{ge}_λ respectively. After this the algorithm also moves to the preceding iteration $j-1$, however, now it decreases h^{bt} with the control duration $X^*_{h^{\text{bt}},j}$.

The process of evaluating control decisions $X^*_{h^{\text{bt}},j}$ is repeated until the first iteration $j = 1$ is reached or when there is no backtrack-horizon left ($h^{\text{bt}} = 0$). If the first iteration is reached and the backtrack-horizon is non-zero, i.e. $j = 1 \wedge h^{\text{bt}} > 0$, then during the first part of the planning the right on green is not granted to a particular stage. This means that the control policy 'R' is added to OS, which indicates that the planning is kept open for any signal group te become green, and its duration of h^{bt} is added to OD.

Finally, before CP^{stages}, $CP^{\text{durations}}$, CP^{gs}_{λ} and CP^{ge}_{λ} can be outputted, the algorithm reverses the entries of all vectors (due to backtracking). The full procedure can be found in Algorithm 3.5.

Algorithm 3.5: Backtracking algorithm for obtaining the optimal control policy from $X^*|_{\forall s \in [1,\ldots,h], \forall j}$

Input : FC, h, j^*, N^{stages}, $X^*_{s,j}$
Output: CP^{duration}, CP^{ge}, CP^{gs}, CP^{stages}

set $h^{\text{bt}} = h$
foreach $j \in [j^* - (N^{\text{stages}} - 1), j^* - (N^{\text{stages}} - 1) - 1, \ldots, 2, 1]$ **do**
    if $X^*_{h^{\text{bt}},j} > 0$ **then**
        append the policy that corresponds to stage $\varphi = j \bmod N^{\text{stages}}$ (e.g. 'A'/'B'/'C') to CP^{stages}, append the policy's duration of $X^*_{h^{\text{bt}},j}$ to CP^{duration} and append the green start and end times for all signal groups $\lambda \in FC$ to CP^{gs}_{λ} and CP^{ge}_{λ} respectively
        subtract $X^*_{h^{\text{bt}},j}$ from h^{bt}
    end
    if $h^{\text{bt}} = 0$ **then**
        reverse the order of the elements in vectors CP^{duration}, CP^{stages}, $CP^{\text{gs}}_{\lambda}|_{\forall \lambda \in FC}$ and $CP^{\text{ge}}_{\lambda}|_{\forall \lambda \in FC}$
        break
    end
    if $h^{\text{bt}} > 0 \wedge j = 1$ **then**
        add policy 'R' to CP^{stages} and add the policy's duration of h^{bt} to CP^{duration}
        reverse the order of the elements in vectors CP^{duration}, CP^{stages}, $CP^{\text{gs}}_{\lambda}|_{\forall \lambda \in FC}$ and $CP^{\text{ge}}_{\lambda}|_{\forall \lambda \in FC}$
    end
end

3.6. Time-to-Green and Time-to-Red Predictions

A key element of the PCD algorithm is that it computes time-to-green (TTG) and time-to-red (TTR) predictions for all of the signal groups $\lambda \in FC$. The procedure for obtaining these predictions is explained in this section and can be observed in Algorithm 3.6. The generic notation that is introduced in Tables 3.10 and 3.11 applies.

Variable	*Domain*	*Unit*	*Description*
CP^{ge}_{λ}	-	-	A vector containing the green end times for signal group λ according to the optimal planning of control decisions. Example: $[9, 35]$
CP^{gs}_{λ}	-	-	A vector containing the green start times for signal group λ according to the optimal planning of control decisions. Example: $[4, 28]$
h	$\mathbb{N}_1$	s	Full planning horizon length
k	$\mathbb{N}_1$	s	Time index
$TTG^{\text{certainty}}_{\lambda,k}$	-	-	The time-to-green certainty indication for signal group λ at time k. This certainty indication equals either '=' or '<>', which denotes a certain or uncertain time-to-green prediction respectively and which depends on whether or not signal group λ is included in the planning
$TTG^{\text{penalty}}_{\lambda,k}$	$\mathbb{N}_0$	s	The summed change of the time-to-green prediction for signal group λ at time k. A change results from a new planning being determined in which signal group λ is planned to become green at a different moment in time

Table 3.10: A list of variables, their domains, their units and their descriptions.

Variable	*Domain*	*Unit*	*Description*
$TTG^{\text{time}}_{\lambda,k}$	$\mathbb{N}_0$	s	The time-to-green prediction for signal group λ at time k
$TTR^{\text{certainty}}_{\lambda,k}$	-	-	The time-to-red certainty indication for signal group λ at time k. This certainty indication equals either '=' or '<>', which denotes a certain or uncertain time-to-green prediction respectively and which depends on whether or not signal group λ is included in the planning
$TTR^{\text{penalty}}_{\lambda,k}$	$\mathbb{N}_0$	s	The summed change of the time-to-red prediction for signal group λ at time k. A change results from a new planning being determined in which signal group λ is planned to become red at a different moment in time
$TTR^{\text{time}}_{\lambda,k}$	$\mathbb{N}_0$	s	The time-to-red prediction for signal group λ at time k

Table 3.11: A list of variables, their domains, their units and their descriptions.

Algorithm 3.6: The algorithm for determining time-to-green and time-to-red predictions for all of the signal groups $\lambda \in FC$.

Input : CP^{ge}_{λ}, CP^{gs}_{λ}, FC, h

Output: $TTG^{\text{certainty}}_{\lambda,k}$, $TTG^{\text{penalty}}_{\lambda,k}$, $TTG^{\text{time}}_{\lambda,k}$, $TTR^{\text{certainty}}_{\lambda,k}$, $TTR^{\text{penalty}}_{\lambda,k}$, $TTR^{\text{time}}_{\lambda,k}$

foreach $\lambda \in FC$ **do**
    set $TTG^{\text{time}}_{\lambda,k} = -999$, $TTG^{\text{certainty}}_{\lambda,k} =$ '<>', $TTR^{\text{time}}_{\lambda,k} = -999$ and $TTR^{\text{certainty}}_{\lambda,k} =$ '<>'
    foreach *policy* $\in CP^{\text{gs}}_{\lambda}$ **do**
        if $CP^{\text{ge}}_{\lambda} \geq 0$ **then**
            if $TTG^{\text{time}}_{\lambda,k} = -999 \vee TTG^{\text{time}}_{\lambda,k} = CP^{\text{gs}}_{\lambda}$ **then**
                set $TTG^{\text{time}}_{\lambda,k} = CP^{\text{gs}}_{\lambda}$, $TTG^{\text{certainty}}_{\lambda,k} =$ '=', $TTR^{\text{time}}_{\lambda,k} = CP^{\text{ge}}_{\lambda}$ and $TTR^{\text{certainty}}_{\lambda,k} =$ '='
                if $CP^{\text{ge}}_{\lambda} = h$ **then**
                    set $TTR^{\text{certainty}}_{\lambda,k} =$ '<>'
                end
            end
        end
    end
    if $TTG^{\text{time}}_{\lambda,k} = -999$ **then**
        set $TTG^{\text{time}}_{\lambda,k} = h$, $TTG^{\text{certainty}}_{\lambda,k} =$ '<>', $TTR^{\text{time}}_{\lambda,k} = 0$ and $TTR^{\text{certainty}}_{\lambda,k} =$ '='
    end
    if $TTG^{\text{time}}_{\lambda,k-1} \neq TTG^{\text{time}}_{\lambda,k}$ **then**
        $TTG^{\text{penalty}}_{\lambda,k} = TTG^{\text{penalty}}_{\lambda,k-1} + (TTG^{\text{time}}_{\lambda,k-1} - TTG^{\text{time}}_{\lambda,k})$
    end
    if $TTR^{\text{time}}_{\lambda,k-1} \neq TTR^{\text{time}}_{\lambda,k}$ **then**
        $TTR^{\text{penalty}}_{\lambda,k} = TTR^{\text{penalty}}_{\lambda,k-1} + (TTR^{\text{time}}_{\lambda,k-1} - TTR^{\text{time}}_{\lambda,k})$
    end
end

Algorithm 3.6 starts by setting the TTG and TTR (TTR) times to the default value of −999 and setting the TTG and TTR certainty indication to '<>'. Considering a signal group λ, the algorithm evaluates all of the policies in CP^{gs}_{λ} and CP^{ge}_{λ}.

If the green end time is at least equal to zero ($CP^{\text{ge}}_{\lambda} \geq 0$), then signal group λ is granted the right on green during the planning, and hence TTG and TTR times can be predicted. A TTG and TTR prediction for signal group λ is determined if no prediction has yet been set ($TTG^{\text{time}}_{\lambda,k} = -999$), or when the green end time increases while the green start time remains unchanged ($TTG^{\text{time}}_{\lambda,k} = GS_{\lambda,\text{policy}}$). This latter situation occurs if signal group λ is granted the right on green in two or more successive stages.

If all of the policies in CP^{gs}_{λ} and CP^{ge}_{λ} have been evaluated and if no TTG and TTR prediction is set for signal group λ, then no green time is allocated to signal group λ in the planning of control decisions and hence the TTG prediction equals at least the planning horizon h. In this case the certainty indication is set to uncertain. With respect to the TTR prediction, the signal(s) of signal group λ are red and remain red, and hence the TTR prediction equals zero.

As a final step, the TTG and TTR penalty is determined for each signal group $\lambda \in FC$ to - if necessary - include a penalty if a signal group λ had a TTG/TTR prediction with certainty indication '=' in the previous time step $k-1$. A penalty is not added for signal groups $\lambda \in FC$ that have an uncertain TTG/TTR predictions, since the accuracy of the prediction is then greatly dependent on the penalty that is used for this case.

3.7. Computation Time

Since the computation time is an important topic in the development of the PCD algorithm, a separate section is dedicated to this subject. Table 3.12 includes the computation times of the PCD algorithm for different numbers of stages and horizon intervals. Figure 3.21 also displays this data, but then in a visual form. Due to large colour interval sizes in this figure (25 seconds), the increase of the computation time for smaller horizon lengths is not properly displayed, which is why Figure 3.20 is also included in this thesis.

90	49.236	75.413	91.22	126.195	153.219	144.622	162.688	173.061
80	30.61	48.42	61.645	81.418	103.221	95.163	107.361	124.981
70	19.821	28.205	38.585	52.81	66.628	63.824	68.909	79.022
60	11.492	16.142	24.26	30.544	36.781	37.278	41.574	44.271
50	5.574	8.228	13.079	16.654	20.326	20.433	23.965	26.028
40	3.208	4.724	6.038	7.464	8.631	10.287	11.442	12.523
35	2.161	2.911	3.83	5.026	5.805	6.975	7.787	8.662
30	1.304	1.766	2.363	2.9	3.331	4.089	4.51	4.951
25	0.664	0.891	1.193	1.48	1.745	2.333	2.592	2.735
20	0.288	0.399	0.518	0.711	0.803	1.079	1.202	1.225
15	0.104	0.159	0.205	0.283	0.342	0.401	0.443	0.478
10	0.025	0.033	0.042	0.064	0.089	0.11	0.12	0.135
	2	3	4	5	6	7	8	9

Table 3.12: Computation time in seconds of the PCD algorithm for different numbers of stages and horizons intervals. On the x-axis the number of stages is represented and the y-axis represents the number of horizon intervals. The horizon intervals are of equal length and represent one second.

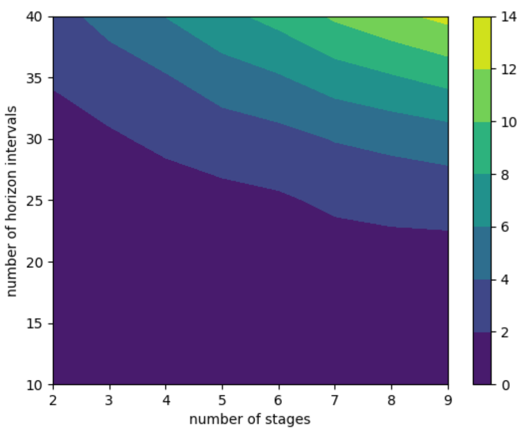


Figure 3.20: Visual representation of the computation times of the PCD algorithm (see Table 3.12).

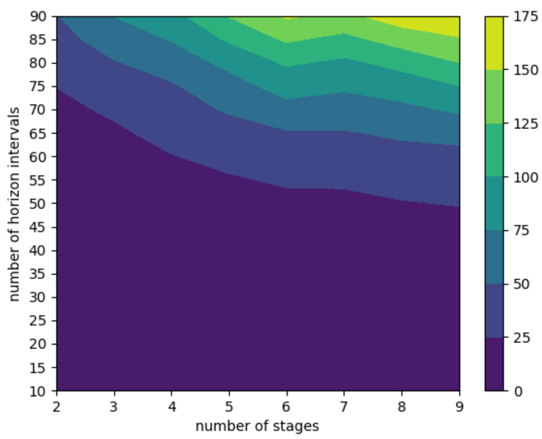


Figure 3.21: Visual representation of the computation times of the PCD algorithm (see Table 3.12).

Please note that the computation times that are presented in Table 3.12 and Figures 3.20 and 3.21 differ for each device on which the algorithm is run. For this study the simulation runs have been executed on a device with an Intel i7-7700 processor and 64 GB of internal memory, equipped with a 64-bit version of Windows 10. Computation times could even differ with respect to the programming language that is used to write the algorithm's code in (the PCD algorithm is written in Python).

As can be observed from Table 3.12 and Figures 3.20 and 3.21, computation times are fairly high. Several combinations of stages and number of intervals are even infeasible (see the red values), since the computation time outgrows the planning horizon. From the tables and the figures it is obvious that the PCD algorithm is unable to run in real time if a planning horizon is considered that exceeds some 10 seconds.

The tables and figures show that the computation time increases if the number of stages and the number of horizon intervals increases. According to the figures, an increase of the horizon has the biggest impact on the computation time, since the lines in the figures are more horizontal than vertical. A quantitative analysis of Table 3.12 supports this. That is, each time the number of stages doubles, the computation time increases with a factor ≈ 1.9, while each time the horizon doubles the computation time seems to increase with a factor ≈ 10. Computation time of the PCD algorithm also increases with an increasing size of the intersection. That is, the larger the number of signal groups $\lambda \in FC$, the larger the computation time.

Given that the planning of control decisions is potentially to be recalculated a number of times per minute, the maximum allowed computation time lays between 5 and 10 seconds (probably closer to 5). As a result of that, the PCD algorithm should be capped somewhere along the line of 30 intervals and 9 stages and 40 intervals and 4 stages (see Table 3.12 and Figure 3.20). Although this cap is sufficient in terms of delay reduction, it was found that optimality in control is not particularly sensitive to vehicle arrivals beyond 25 seconds in the future, this cap does not necessarily comply with the desire of Goudappel Coffeng to plan for a long horizon (preferably 2 minutes or 120 seconds, thus 120 intervals).

3.8. Summary

The main findings of this chapter are:

- Next to increasing the predictability of control decisions, the goal of the new traffic signal control algorithm is to reduce the delay and the number of stops that vehicles experience by better anticipating onto arriving vehicles. To do so, a *vehicle arrival* is to be defined in a way that it supports this thought. By defining a vehicle arrival to be the moment in time at which a vehicle arrives at the stop line of a signal group, this support is achieved;
- The PCD algorithm that is developed in this chapter is a dynamic programming formulation that adapts the basic forward recursion optimisation technique that is introduced by Sen and Head (1997) and is also applied by van Katwijk (2008) and Chen and Sun (2016). This technique splits the full planning horizon up into smaller, more manageable parts for which it calculates control policies, after which it constructs an overall control policy by combining the best performing control policies for these smaller parts. Using this technique, it can be prevented that all control options need to be evaluated, which saves a lot of computation time. The PCD algorithm cannot guarantee that the overall optimal solution is found;
- Like the algorithm by van Katwijk (2008), the PCD algorithm considers a signal-group based approach in which ahead of time realisations of signal groups is incorporated. This approach better represents Dutch actuated/adaptive traffic signal control than stage-based approaches such as the ones by Sen and Head (1997) and Chen and Sun (2016). In comparison to a stage-based approach, a signal-group based approach allows for more accurate time-to-green and time-to-red predictions. Like the latter two algorithms, the PCD algorithm considers stages over blocks in which signal groups can be present multiple times;
- Prioritisation of vehicles, one of the key aspects of Talking Traffic, is directly implemented in the PCD algorithm. This is made possible by keeping track of the composition of a queue. By incorporating weight factors on the level of (a) individual vehicles, (b) vehicle types and (c) signal groups specific road users can be prioritised over others. Currently, prioritisation is mainly intended for public transport and emergency vehicles, but one could think about prioritising heavy trucks over other road users in the future. As long as the vehicle that is to be treated with priority can be detected, the PCD algorithm is able to incorporate its priority;

- Theoretically the PCD algorithm is able to handle even the smallest thinkable time intervals. However, in say a 40 second horizon, using small time intervals would greatly widen the algorithm's search space, which results in the computation time growing far beyond acceptable boundaries. Hence, choosing a feasible number of intervals over which optimisation is to take place is very important. Considering that a planning is potentially to be determined multiple times a minute, the computation time may not exceed a couple of seconds, which means that 9 stages and 25 intervals or 4 stages and 40 intervals are to be considered. Due to the magnitude of the computation times, real time execution of the PCD algorithm is unthinkable without using devices with excessive computational capacity.

4

Traffic Signal Control Algorithm

This chapter describes a newly developed traffic signal control (TSC) algorithm. The algorithm uses the output from the PCD algorithm as a framework to grant the right on green to signal groups and it determines the colour of all signals for the current time instant. Section 4.1 begins the chapter by describing the need for the existence of a TSC algorithm parallel to the PCD algorithm. This need is twofold and relates to (a) computation time and (b) inaccurate assumptions while creating a planning of control decisions. Section 4.2 discusses the TSC algorithm's set-up. This section elaborates on how the TSC algorithm functions and how the planning by the PCD algorithm is adapted to actual traffic conditions. Section 4.3 provides a summary of the chapter.

As has become clear from the literature review of Chapter 2, algorithms that adjust a static planning of control decisions to actual traffic conditions are far from widely discussed. Hence, many of the concepts that are described in this chapter, especially the ones that relate to determining which signal groups are allowed to become green, are newly developed.

4.1. The Need for a TSC Algorithm

Typical Dutch traffic signal control algorithms allow control changes to take place every tenth of a second. Each time these algorithms recalculate a wide range of functions is run that, among other things, determine detector occupancies and signal colours (red/amber/green). Such recalculation rates allow the colour of signals to potentially change every tenth of a second (e.g. a signal may change to amber only 0.1 second after the extension loop detector has become unoccupied) and hence there is low time loss. Given that the PCD algorithm has a computation time that is far larger than a tenth of a second (see Section 3.7), the PCD algorithm itself does not allow for similar efficiency if code were to be included to determine signal colours.

Next to the computation time, there is another reason for requiring a TSC algorithm additional to the PCD algorithm. That is, in order to create a planning of control decisions some assumptions are made that by definition contain (small) errors and as a result strictly executing the planning of control decisions is inefficient from the point of view of delay reduction. Three examples errors are:

- Queue dissolvement depends on driving behaviour of road users. Some road users have shorter reaction times and/or accelerate more aggressively than others, which means that by definition the rate at which queues dissolve in practice does not precisely equal the queue discharge rate that is considered in the PCD algorithm. Although the exact difference between the actual and assumed saturation flow is prone to fluctuation and will likely be unknown, the difference will typically stay within a 10% range (e.g. assumed: 2000 veh/h, actual: 2150 veh/h);
- Prediction of vehicle arrivals depends on extrapolation, either linear as in the Run and Repeat method or via more complicated and accurate car-following models, and vehicles will very likely not arrive at the exact time that they are predicted to arrive;
- In case vehicle arrivals are not available, the PCD algorithm estimates the expected number of arrivals on the basis of the current traffic flow, which may result in under- or overestimation of traffic presence.

The idea of the TSC algorithm is that it adapts the output by the PCD algorithm to bridge the gap between the static planning on the one hand and the dynamic actual traffic conditions on the other hand. By doing so, inefficiency of control decisions that follows from incorrect assumptions is reduced.

4.2. Algorithm Set-up and Functionalities

Like the Golden Controller, the TSC algorithm that is developed in this study applies a variable order in which signal groups are given green. It thereby differs greatly from the standard Dutch practice of traffic signal control. The TSC algorithm however also maintains a number of standard procedures, such as using loop detectors to end and extend green and the application of signal completion as described in Appendix A.

Like typical Dutch traffic signal control algorithms, the TSC algorithm recalculates every 0.1 second. Each time the TSC algorithm recalculates, the algorithm basically executes four steps:

- First, traffic signal control inputs are obtained and various internal elements are updated on the basis of this. That is, detector occupancies are retrieved for all detectors and detector gap times are updated accordingly. Green requests are determined for all signal groups and using the distant and stop line loop detectors vehicles are added and/or removed from the queues. This first step of the TSC algorithm, of which the activities are elaborated on in detail in Section 4.2.1, also includes determining whether or not the planning of control decisions is to be recalculated;
- The activities of the second step of the TSC algorithm, which are described in detail in Section 4.2.2, are only executed when it has been decided that the planning of control decisions is to be recalculated. During this second step all of the input-data for the PCD algorithm is established. That is, the queues that are present at the current time step are set to be the initial queues and the green end times for all signal groups are determined. The green end times represent the prior commitments of the control policy. For example, if at the moment of recalculation of the planning the signals of a signal group are green but have not shown green for the duration of at least the fixed green time, then the PCD algorithm is required to take the remaining green time of that signal group into account when making a new planning of control decisions. The expected vehicle arrivals over the planning horizon are also determined during this second step of the TSC algorithm.

  After a new planning of control decisions is determined by the PCD algorithm, the output that is generated needs to be processed. That is, the sequence of stages (CP^{stages}) needs to be translated into the right on green for individual signal groups and the corresponding green duration needs to be derived. All of these activities, referred to as *green allowance* in this thesis, also take place during the second step of the TSC algorithm;
- The third step of the TSC algorithm includes the signal completion part. During this step, which is described in detail in Section 4.2.3, the right on green for signal groups as determined in the second step of the TSC algorithm is translated into actual signal colours by walking through all of the components of signal completion in successive order;
- The fourth and final step of the TSC algorithm (see Section 4.2.4) includes updating various internal timers that are relevant for proper working of the TSC algorithm. That is, on the basis of the signal colours that were determined during the signal completion process, signal durations are updated. Also, various timers which include the waiting times and time-to-green and time-to-red predictions are updated.

The generic notation of variables that is introduced in Tables 4.1, 4.2 and 4.3 applies for the mathematical description of all of the activities of the four steps of the TSC algorithm.

Variable	*Domain*	*Unit*	*Description*
$A_{\lambda,k}$	$\mathbb{N}_0$	veh	The number of vehicles that are expected to arrive at signal group λ at time k
$\alpha_{\lambda,k}$	-	-	This boolean denotes whether or not signal group λ has a green request at time k
$\beta_{\lambda,k}$	-	-	This boolean denotes whether or not signal group λ is blocked from becoming green at time k
$c(\lambda,k)$	-	-	The queue composition of signal group λ at time k. Each of the entries in this vector represents the type of vehicle. Example: $[Car, Car, Bus, Car]$

Table 4.1: A list of variables, their domains, their units and their descriptions.

Variable	*Domain*	*Unit*	*Description*
CP^{duration}	-	time steps	A vector containing the optimal duration of stages. Example: $[8,13,5]$
CP^{stages}	-	-	A vector containing the optimal sequence of stages. Example: $[B,A,C]$
$\gamma_{\lambda,k,v}$	-	-	This boolean denotes whether or not detector v of signal group λ is occupied at time k
FC	-	-	A user-defined ordered set of signal groups
FC^{rog}	-	-	A set of signal groups $\lambda \in FC$ that are included in the planning of control decisions. The set is ordered according to the green start time of signal groups
k	$\mathbb{N}_1$	s	Time index
λ	$\mathbb{N}_1$	-	Signal group index
$q(s,x,j,\lambda,k)$	-	-	The queue fraction of signal group λ at time k. Each of the entries in this vector represents the fraction of a vehicle that is still present in the queue, which by definition equals 1 in case of TSC algorithm. Example: $[1,1,1,1]$
$s^{\text{colour}}_{\lambda,k}$	-	-	The colour that the signals of signal group λ display to road users at time k. This colour can either be *red*, *amber* or *green*
$s^{\text{status}}_{\lambda,k}$	-	-	The internal status of signal group λ at time k. This status can either be *RV*, *RA*, *FG*, *VG*, *MG* or *GL*
t^{fg}_{λ}	$\mathbb{N}_1$	s	The fixed green time of signal group λ. The fixed green time represents the minimum green time
t^{ga}_{λ}	$\mathbb{N}_0$	s	The time that signal group λ may be given green additional to the green time that is incorporated for signal group λ in the planning of control decisions
$t^{\text{gap}}_{\lambda,k,v}$	$\mathbb{R}$	s	The gap time for detector v of signal group λ at time k. The gap time represents the elapsed time since the detector was last occupied
$t^{\text{gap}^{\text{set}}}_{\lambda,k,d}$	$\mathbb{R}$	s	The set gap time for detector d of signal group λ at time k. The set gap time represents the time that a detector may be unoccupied before any action such as ending green may take place
$t^{\text{gd}}_{\lambda,k}$	$\mathbb{R}$	s	The elapsed time at time k since the signal(s) of signal group λ turned green. This value is zero if the signal(s) of signal group λ are currently not green
$t^{\text{gp}}_{\lambda,k}$	$\mathbb{R}$	s	The remaining green time for signal group λ at time k according to the planning of control decisions
$t^{\text{rd}}_{\lambda,k}$	$\mathbb{R}$	s	The elapsed time at time k since the signal(s) of signal group λ turned red. This value is zero if the signal(s) of signal group λ are currently not red
$t^{\text{recalculate}}_{k}$	$\mathbb{R}$	s	The remaining time at time step k before a new planning of control decisions is determined
t^{rg}_{λ}	$\mathbb{N}_1$	s	The guaranteed red time of signal group λ
t^{vg1}_{λ}	$\mathbb{R}$	s	The maximum extension green time of signal group λ on the basis of the stop line, extension and distant loop detectors
t^{vg2}_{λ}	$\mathbb{R}$	s	The maximum extension green time of signal group λ on the basis of the extension and distant loop detectors
$t^{\text{wa}}_{\lambda,k}$	$\mathbb{R}$	s	The waiting time of signal group λ at time k, hence the elapsed time since the signal(s) of signal group λ turned red while there has been a green request. This value is zero if the signal(s) of signal group λ are currently not red
t^{wm}_{λ}	$\mathbb{N}_1$	s	The maximum allowed waiting time of signal group λ

Table 4.2: A list of variables, their domains, their units and their descriptions.

Variable	*Domain*	*Unit*	*Description*
t^{y}_{λ}	$\mathbb{N}_1$	s	The set amber time of signal group λ
$t^{\mathrm{yd}}_{\lambda,k}$	$\mathbb{R}$	s	The elapsed time at time k since the signal(s) of signal group λ turned amber. This value is zero if the signal(s) of signal group λ are currently not amber
υ	-	-	Detector index
$V^{\mathrm{type}}_{\lambda,k}$	-	-	The type of vehicle that occupies the distant loop detector of signal group λ at time k
$V^{\mathrm{weight}}_{\lambda,k}$	$\mathbb{R}$	-	The weight of the vehicle that occupies the distant loop detector of signal group λ at time k
$w(\lambda,k)$	-	-	The queue weight of signal group λ at time k. Each of the entries in this vector represents the weight of a vehicle. Example: $[1,1,3,1]$
$\omega^{\mathrm{distance}}_{\lambda}$	$\mathbb{N}_1$	meter	The distance between the distant loop detector and the stop line of signal group λ
$\omega^{\mathrm{velocity}}_{\lambda}$	$\mathbb{N}_1$	km/h	The average velocity with which vehicles at signal group λ typically travel between the distant loop detector and the stop line

Table 4.3: A list of variables, their domains, their units and their descriptions.

4.2.1. Updating Traffic Signal Control Inputs

At the start of recalculation of the TSC algorithm a number of activities are performed that update the inputs of the traffic signal control algorithm. These activities are explained in detail in the next paragraphs.

Obtaining Detector Occupancies
For each signal group $\lambda \in FC$ the detector occupancy $\gamma_{\lambda,k,\upsilon}$ is obtained for all of its detectors υ at the current time instant k. This detector status can either be 0 or 1, denoting the detector is unoccupied or occupied respectively. The following mathematical definition is applied:

$$\gamma_{\lambda,k,\upsilon} \in \{0,1\} \qquad \forall \lambda \in FC, \quad \forall \upsilon \tag{4.1}$$

Updating Detector Gap Times
Using the equation below the detector gap time $t^{\mathrm{gap}}_{\lambda,k,\upsilon}$ is determined for each of the detectors υ of signal group $\lambda \in FC$ at time k. The detector gap time is increased by one decisecond if the detector is unoccupied and the detector gap time is reset to zero if the detector is occupied.

$$t^{\mathrm{gap}}_{\lambda,k,\upsilon} = \begin{cases} 0 & \text{if } \gamma_{\lambda,k,\upsilon} = 1 \\ t^{\mathrm{gap}}_{\lambda,k-1,\upsilon} + 0.1 & \text{if } \gamma_{\lambda,k,\upsilon} = 0 \end{cases} \qquad \forall \lambda \in FC, \quad \forall \upsilon \tag{4.2}$$

Determining Green Requests
For each signal group $\lambda \in FC$ its green request $\alpha_{\lambda,k}$ at time k is determined via:

$$\alpha_{\lambda,k} = \begin{cases} \mathit{True} & \text{if } \max(\gamma_{\lambda,k,1},\ldots,\gamma_{\lambda,k,n}) = 1 \land t^{\mathrm{rd}}_{\lambda,k} \geq t^{\mathrm{rg}}_{\lambda} \\ \mathit{False} & \text{otherwise} \end{cases} \qquad \forall \lambda \in FC \tag{4.3}$$

In this equation a green request for signal group λ is created when at least one of the detectors υ is occupied, while the signal has shown red for at least the duration of the guaranteed red time. The latter constraint is introduced so that is prevented that vehicles that pass over the stop line loop detector at the first moment(s) of red create a green request.

Updating Queues
One of two main inputs of the PCD algorithm are the queues at each of the signal groups $\lambda \in FC$. To determine these queues, the distant and stop line loop detectors are used. The procedures that are described in the enumeration and the equations below are applied in the TSC algorithm. These procedures are in accordance with the procedures that the Golden Controller applies.

- The queue of signal group λ at time k is increased with one vehicle if the distant loop detector is occupied in the current time step k and if that same distant loop detector was unoccupied in the previous time step $k-1$;
- The queue of signal group λ at time k is decreased with one vehicle if the stop line loop detector is unoccupied in the current time step k and if that same stop line loop detector was occupied in the previous time step $k-1$.

$$
\begin{aligned}
c(\lambda,k) &= \begin{cases} [c(\lambda,k-1),[V^{\text{type}}_{\lambda,k}]] \text{ if } \gamma_{\lambda,k,dd}=1 \wedge \gamma_{\lambda,k-1,dd}=0 \\ c(\lambda,k-1)[1::] \text{ if } \gamma_{\lambda,k,ds}=0 \wedge \gamma_{\lambda,k-1,ds}=1 \end{cases} && \forall \lambda \in FC \\
q(\lambda,k) &= \begin{cases} [q(\lambda,k-1),[1]] \text{ if } \gamma_{\lambda,k,dd}=1 \wedge \gamma_{\lambda,k-1,dd}=0 \\ q(\lambda,k-1)[1::] \text{ if } \gamma_{\lambda,k,ds}=0 \wedge \gamma_{\lambda,k-1,ds}=1 \end{cases} && \forall \lambda \in FC \\
w(\lambda,k) &= \begin{cases} [w(\lambda,k-1),[V^{\text{weight}}_{\lambda,k}]] \text{ if } \gamma_{\lambda,k,dd}=1 \wedge \gamma_{\lambda,k-1,dd}=0 \\ w(\lambda,k-1)[1::] \text{ if } \gamma_{\lambda,k,ds}=0 \wedge \gamma_{\lambda,k-1,ds}=1 \end{cases} && \forall \lambda \in FC \\
&&& \text{where:} \\
&&& dd = \text{distant loop detector} \\
&&& ds = \text{stop line loop detector}
\end{aligned}
\tag{4.4}
$$

When counting vehicles with loop detectors one has to pay attention to the cumulative error problem (van Erp et al., 2018). The cumulative error problem arises when loop detectors miss vehicles or count vehicles twice, for instance because of measurement inaccuracies or when a vehicle occupies two detectors when it changes lanes. This error results in too few or too many vehicles being added or subtracted from a queue, which needs to be corrected for. To prevent that the cumulative error problem occurs in the TSC algorithm, the queues c, q and w for signal group λ are reset when signal group λ has no green request ($\alpha_{\lambda,k} = False$) and when neither of its detectors are occupied ($\max(\{\gamma_{\lambda,k,1},\ldots,\gamma_{\lambda,k,n}\}) = 0$). The following equation describes this in mathematical terms:

$$
\begin{aligned}
c(\lambda,k) &= \emptyset \text{ if } \alpha_{\lambda,k} = False \wedge \max(\{\gamma_{\lambda,k,1},\ldots,\gamma_{\lambda,k,n}\}) = 0 && \forall \lambda \in FC \\
q(\lambda,k) &= \emptyset \text{ if } \alpha_{\lambda,k} = False \wedge \max(\{\gamma_{\lambda,k,1},\ldots,\gamma_{\lambda,k,n}\}) = 0 && \forall \lambda \in FC \\
w(\lambda,k) &= \emptyset \text{ if } \alpha_{\lambda,k} = False \wedge \max(\{\gamma_{\lambda,k,1},\ldots,\gamma_{\lambda,k,n}\}) = 0 && \forall \lambda \in FC
\end{aligned}
\tag{4.5}
$$

Determining the Need for Recalculation of the Planning of Control Decisions
In the PCD and TSC algorithms signal groups are considered as individual units. In practice this means that at any time k signal groups are typically in different internal states (e.g. FG, VG and MG. See Appendix A). Due to this, defining the right moment to recalculate the planning of control decisions is without a doubt one of the most difficult things in look-ahead traffic signal control. Unlike in a stage-based approach where recalculation could take place during the fixed green time (FG) of the signal groups[1], there is not one particular moment that offers great opportunity for recalculation of the control policy in a signal-group based approach. Although this is no problem in a simulation environment, microsimulation models can rather easily be paused to recalculate the planning of control decisions, a traffic signal controller on the street can obviously not just be paused for several seconds.

For simplicity reasons it is decided to neglect the above mentioned difficulty in this study and allow recalculations to take place at any point in time, hence also at times that would be inconvenient in practice. Picking a good moment to recalculate, or possibly taking a whole different approach by continuously determining an optimal planning in a computation process that runs parallel to the traffic signal controller, is however an important topic for future work. The TSC algorithm recalculates in either one of the following situations:

[1] During this period the signal colours are fixed for several seconds, no matter of the presence of traffic, and hence there is sufficient time available to recalculate the optimal control policy.

- The right on green is not dedicated to a particular stage while at least one signal group has a green request ($CP^{\text{stages}} = \emptyset \wedge True \in \{\alpha_{\lambda,k} : \forall \lambda \in FC\}$).
- Each of the signal groups that is included in the first stage of the optimal sequence of stages (CP^{stages}) is blocked, denoting that each signal group either has no green request or has been green and is now red again or about to turn red ($\beta_{\lambda,k} = True \, \forall \lambda \in CP_1^{\text{stages}}$);
- The remaining time until recalculation of the optimal control policy has elapsed ($t^{\text{recalculate}} < 0$);

The first situation applies to the time where the planning does not include a particular stage (at the time of recalculation there were no queues and no expected vehicle arrivals), while there is now a green request. The second condition makes sure that the planning of control decisions is recalculated when all of the signal groups in the first stage of the control policy are given green, hence when the first policy of CP^{stages} has fully been executed. The third condition is introduced so that signal groups may potentially be given green for a time longer than the planning horizon. This is especially relevant if short planning horizons are considered.

4.2.2. Recalculating the Planning of Control Decisions

If it has been decided by the TSC algorithm to recalculate the planning of control decisions, then the TSC algorithm starts the process of gathering all of the data that the PCD algorithm requires as an input. During this process the following activities are performed.

Obtaining Current Queues

The first part of gathering the input-data for the PCD algorithm consists of obtaining the current queues c, q and w for each of the signal groups $\lambda \in FC$. These queues are represented by the queues that are present in the current time step k and hence the following equation applies:

$$\begin{aligned}
c_\lambda^{\text{initial}} &= c(\lambda, k) && \forall \lambda \in FC \\
q_\lambda^{\text{initial}} &= q(\lambda, k) && \forall \lambda \in FC \\
w_\lambda^{\text{initial}} &= w(\lambda, k) && \forall \lambda \in FC
\end{aligned} \tag{4.6}$$

Obtaining Expected Vehicle Arrivals

The second part of gathering the input-data for the PCD algorithm is obtaining the expected vehicle arrivals over the planning horizon h. In doing this, either one of the following situations is the case for any $\lambda \in FC$:

- Vehicle arrivals are available;
- Vehicle arrivals are not available or there is a too large mismatch between the number of expected vehicle arrivals on the one hand and the current traffic flow on the other hand.

In the first situation the vehicle arrivals can directly be inputted into the PCD algorithm. The only modification to perform is setting the vehicle arrivals of the first couple of seconds to zero, so that it is prevented that vehicles are double counted. Vehicles that arrive in the first couple of seconds have namely already been counted by the distant loop detector. The following equation is applied to determine exactly over how many seconds vehicle arrivals need to be reset to zero:

$$A_{\lambda,k} = 0 \text{ if } k \leq \omega_\lambda^{\text{distance}} / (\omega_\lambda^{\text{velocity}} / 3.6) \qquad \forall \lambda \in FC \tag{4.7}$$

The second situation is more complex since this situation requires vehicle arrivals to be added manually. Vehicle arrivals are added with a rate that is equal to the current traffic flow so that it is prevented that the PCD algorithm incorporates too few green time.

Determining Green End Times

The third and final part of gathering the input-data for the PCD algorithm consists of obtaining the green end times for all signal groups $\lambda \in FC$. These green end times are determined via:

$$ge_\lambda^{\text{initial}} = \begin{cases} -1 * (t_{\lambda,k}^{\text{rd}} + t_\lambda^{\text{y}}) & \text{if } t_{\lambda,k}^{\text{rd}} > 0 \\ -1 * t_{\lambda,k}^{\text{yd}} & \text{if } t_{\lambda,k}^{\text{yd}} > 0 \\ \max(0, (t_\lambda^{\text{fg}} - t_{\lambda,k}^{\text{gd}})) & \text{if } t_{\lambda,k}^{\text{gd}} > 0 \end{cases} \qquad \forall \lambda \in FC \tag{4.8}$$

Equation 4.8 states that when the signal is currently red ($t^{\text{rd}}_{\lambda,k} > 0$), that the green end time equals the negative sum of the current red duration plus the set amber time. If the signal is currently amber ($t^{\text{yd}}_{\lambda,k} > 0$), then the green end time equals the negative elapsed amber time. Finally, if the signal is currently green ($t^{\text{gd}}_{\lambda,k} > 0$), then the green end time is the maximum of 0 and the fixed green time minus the elapsed green time. Please note that the green end times are calculated from the point of view of the current time instant, hence if the signal of signal group λ is currently red, then the green end time of λ is a negative value.

Recalculating the Planning of Control Decisions
With all of the input-data gathered, a new planning of control decisions can now be determined. For this purpose the PCD algorithm that is explained in Chapter 3 is executed.

Green Allowance
After a new planning of control decisions has been determined by the PCD algorithm, the next step is to translate this planning into signal colours. To do so, two processes need to be executed:

- A *higher level* process that translates the control policy into the order in which signal groups may become green, thereby effectively determining the right on green for all signal groups $\lambda \in FC$. This higher level process is executed only once, namely after a new planning has been determined, and it is discussed in detail in this paragraph. For all of the signal groups $\lambda \in FC$ the green time also needs to be derived from the planning of control decisions, which naturally is 0 for signal groups that are not granted the right on green in the planning;
- A *lower level* process, executed every tenth of a second for as long as the TSC algorithm runs, checks (and adjusts) the validity of the right on green for all signal groups $\lambda \in FC$ and translates this right on green to signal colours. This lower level process is performed by the signal completion part of the TSC algorithm and is explained in Section 4.2.3.

To facilitate the higher level process, the following variables are defined:

- CP^{stages} denotes the order in which stages are granted the right on green and is directly obtained from the PCD algorithm. CP^{stages} includes stages $\varphi \in SG$, but does not necessarily have to include all of the stages in SG. Stages can also be present in CP^{stages} multiple times, but never successively;
- FC^{rog} is a derivative of OD and denotes the order in which signal groups are granted the right on green in the planning of control decisions. This order is derived from the the time-to-green and time-to-red predictions via Algorithm 4.1, which itself is a derivative of the green start and end times (see Section 3.7). FC^{rog} includes signal groups $\lambda \in FC$, but does not necessarily have to include all of the signal groups in FC. Signal groups can only be present once in FC^{rog}. Signal groups that are not granted the right on green in the planning are not present in FC^{rog};

Algorithm 4.1: The algorithm for deriving FC^{rog} from predicted time-to-green times.

```
Input  : FC, h, TTG^time
Output : FC^rog
set FC^rog = ∅ and FC* = argsort(TTG^time_{∀λ,t})
foreach λ ∈ FC* do
    if TTG^time_{λ,t} ≠ h then
        FC^rog = [FC^rog, λ]
    end
end
```

- $t^{\text{gp}}_{\lambda,k}$ denotes the amount of time that signal group λ is allowed to be green according to the optimal planning. This green time is calculated as the time-to-red prediction minus the time-to-green prediction for signal groups that are currently amber or red and in FC^{rog}, and as the time-to-red prediction for signal groups that are currently green and in FC^{rog} (see Equation 4.9). If signal group $\lambda \notin FC^{\text{rog}}$, then the allowed green time naturally equals 0;

$$\lambda^{\text{gp}}_{\lambda,k} = \begin{cases} TTR^{\text{time}}_{\lambda,k} & \text{if } \lambda \in FC^{\text{rog}} \wedge s^{\text{colour}}_{\lambda,k} = green \\ TTR^{\text{time}}_{\lambda,k} - TTG^{\text{time}}_{\lambda,k} & \text{if } \lambda \in FC^{\text{rog}} \wedge s^{\text{colour}}_{\lambda,k} = amber \vee red \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

- $\beta_{\lambda,k}$ denotes whether or not a signal group λ is blocked from becoming green at time k. $\beta_{\lambda,k}$ is set to *False* for each signal group $\lambda \in SG^{\text{rog}}$ and it is set to *True* for each signal group $\lambda \notin SG^{\text{rog}}$ via equation:

$$\beta_{\lambda,k} = \begin{cases} False & \text{if } \lambda \in SG^{\text{rog}} \\ True & \text{if } \lambda \notin SG^{\text{rog}} \end{cases} \qquad \forall \lambda \in FC \tag{4.10}$$

- $t^{\text{recalculate}}$ denotes the remaining time before recalculation of the optimal control policy takes place and is set equal to the duration of the first policy in CP^{stages}, hence Equation 4.11 applies.

$$t^{\text{recalculate}} = CP^{\text{stages}}_{1} \tag{4.11}$$

4.2.3. Executing Signal Completion

The lower level process in which the colours of the signals are determined for each of the signal groups $\lambda \in FC$ is performed via signal completion. The signal completion that is included in the TSC algorithm is in the base similar to the one that is described in Appendix A, except that waiting green is left out for simplicity (if desired, this can easily be included). Also, a number of modification are made to allow traffic signal control on the basis of a planning. The TSC algorithm is not limited to the form of signal completion that is described in Appendix A but instead allows for implementation of other forms of signal completion too.

Red Before Request (RV)

The RV state is the default internal state for all signal groups $\lambda \in FC$. During this state it decided whether or not a signal group λ is allowed to enter the signal completion process to become green. If a signal group is allowed to become green then the internal state of that signal group propagates from *RV* to *RA* via:

$$\text{if} \quad s^{\text{status}}_{\lambda,k} \neq RV \vee greenrealisation = True \quad \text{propagate to RA by setting} \quad s^{\text{status}}_{\lambda,k} = RA \tag{4.12}$$

To enter the signal completion process either one the conditions enumerated below must be met. The first condition that is mentioned represents the left part of Equation 4.12 ($s^{\text{status}}_{\lambda,t} \neq RV$) and the second and third condition relate to the right part of this equation (*greenrealisation* = *True*):

- The current internal state of signal group λ does not equal Red Before Request (*RV*), hence $s^{\text{status}}_{\lambda,k} \neq RV$.
- Signal group λ may be realised primary/primary ahead. The following conditions need to be met for this:
  - There is at least one signal group $\lambda^* \in CP^{\text{stages}}_1$ that is not blocked;
  - Signal group λ must have a green request, hence $\alpha_{\lambda,k} = True$;
  - Signal group λ is not blocked ($\beta_{\lambda,k} = False$);
  - All of the conflicting $\lambda^* \in FC$ that proceed λ in CP^{stages} must be blocked.
- Signal group λ may be realised alternatively. The following conditions need to be met for this:
  - There is at least one signal group $\lambda^* \in CP^{\text{stages}}_1$ that is not blocked;
  - Signal group λ must have a green request, hence $\alpha_{\lambda,k} = True$;
  - Signal group λ is blocked ($\beta_{\lambda,k} = True$);
  - All of the conflicting $\lambda^* \in CP^{\text{stages}}$ are either blocked or the predicted time-to-green time of λ^* with certainty restriction '=' is larger than or equal to the sum of:
    - ◇ The fixed green time of signal group λ (t^{fg}_{λ});
    - ◇ The set amber time of signal group λ (t^{y}_{λ});
    - ◇ The clearance time between signal groups λ and λ^* ($t^{\text{cl}}_{\lambda,\lambda^*}$).

Red After Request (RA)
If signal group λ is in the *RA* state it has been decided that that signal group has the right to become green. Now it needs to be checked if it can actually become green too. The internal state of signal group λ propagates to *FG* when both of the conditions that are described by the following equation are met:

$$\text{if} \quad \begin{cases} t^{\mathrm{rd}}_{\lambda,k} \geq t^{\mathrm{rg}}_{\lambda} \\ t^{\mathrm{rd}}_{\lambda^*,k} \geq t^{\mathrm{cl}}_{\lambda^*,\lambda} \quad \forall \lambda^* \in FC \setminus \lambda^* = \lambda \end{cases} \quad \text{propagate to FG by setting} \quad s^{\mathrm{status}}_{\lambda,k} = FG \tag{4.13}$$

The first condition checks if the current red time of signal group λ is larger than or equal to the guaranteed red time of signal group λ and the second condition checks if all of the conflicting signal groups $\lambda^* \in FC$ have at least been red for the duration of the clearance time between λ^* and λ.

Fixed Green (FG)
During the fixed green state green is given for the duration of the fixed green time independent of the presence of traffic. The internal status propagates to extension green (VG) if the condition is met that is described by:

$$\text{if} \quad t^{\mathrm{gd}}_{\lambda,k} \geq t^{\mathrm{fg}}_{\lambda} \quad \text{propagate to VG by setting} \quad s^{\mathrm{status}}_{\lambda,k} = VG \tag{4.14}$$

Extension Green (VG)
During the extension green period green is given as long as a number of conditions are met that mostly rely on the presence of traffic. The internal status propagates from extension green (VG) to prolonging green (MG) according to Algorithm 4.2.

Algorithm 4.2: Algorithm for extending green during the extension green time period (VG)

Input : FC, $t^{\mathrm{fg}}_{\lambda}$, $t^{\mathrm{ga}}_{\lambda}$, $t^{\mathrm{gap}}_{\lambda,k,dd}$, $t^{\mathrm{gap}}_{\lambda,k,de}$, $t^{\mathrm{gap}}_{\lambda,k,ds}$, $t^{\mathrm{gap^{set}}}_{\lambda,k,dd}$, $t^{\mathrm{gap^{set}}}_{\lambda,k,de}$, $t^{\mathrm{gap^{set}}}_{\lambda,k,ds}$, $t^{\mathrm{gd}}_{\lambda,k}$, $t^{\mathrm{gp}}_{\lambda,k}$, $t^{\mathrm{vg1}}_{\lambda}$, $t^{\mathrm{vg2}}_{\lambda}$
Output: $s^{\mathrm{status}}_{\lambda,k}$

foreach $\lambda \in FC$ **do**
  if $t^{\mathrm{gd}}_{\lambda,k} \leq (t^{\mathrm{fg}}_{\lambda} + t^{\mathrm{vg1}}_{\lambda}) \wedge (t^{\mathrm{gap}}_{\lambda,k,ds} \leq t^{\mathrm{gap^{set}}}_{\lambda,ds} \vee t^{\mathrm{gap}}_{\lambda,k,de} \leq \lambda^{\mathrm{gap^{set}}}_{\lambda,de})$ **then**
    continue
  else if $\lambda^{\mathrm{gd}}_{\lambda,k} \leq (t^{\mathrm{fg}}_{\lambda} + t^{\mathrm{vg2}}_{\lambda})$ **then**
    if $t^{\mathrm{gp}}_{\lambda,k} > 0 \wedge (t^{\mathrm{gap}}_{\lambda,kde} \leq t^{\mathrm{gap^{set}}}_{\lambda,de} \vee t^{\mathrm{gap}}_{\lambda,k,dd} \leq t^{\mathrm{gap^{set}}}_{\lambda,dd})$ **then**
      continue
    else if $t^{\mathrm{gp}}_{\lambda,k} > -1 * t^{\mathrm{ga}}_{\lambda} \wedge t^{\mathrm{gap}}_{\lambda,k,de} \leq t^{\mathrm{gap^{set}}}_{\lambda,de}$ **then**
      continue
    else
      propagate to MG by setting $s^{\mathrm{status}}_{\lambda,k} = MG$
    end
  else
    propagate to MG by setting $s^{\mathrm{status}}_{\lambda,k} = MG$
  end
end

As can be observed in Algorithm 4.2, extension green is split up in two parts. In the first part where the current green time $t^{\mathrm{gd}}_{\lambda,k}$ is smaller than or equal to the fixed green time $t^{\mathrm{fg}}_{\lambda}$ plus the first extension green period $t^{\mathrm{vg1}}_{\lambda}$, green extension takes place on the basis of the stop line, extension and distant loop detectors. t^{vg1} is typically limited to a couple of seconds and is there to make sure that vehicles that drive slowly in the first 5 to 10 seconds after the start of green are able to make the green light. During the second part where $t^{\mathrm{gd}}_{\lambda,k} \leq t^{\mathrm{fg}}_{\lambda} + t^{\mathrm{vg2}}_{\lambda}$, extension of green takes place via the extension and distant loop detectors only. After $t^{\mathrm{fg}}_{\lambda} + t^{\mathrm{vg1}}_{\lambda}$ has elapsed it is assumed that the speed of vehicles is sufficient to pass a signal if the signal turns amber some time after the final vehicle in the queue has left the extension loop detector.

In the second part a variable called $t^{\mathrm{ga}}_{\lambda}$ is incorporated. This variable allows signal group λ to be green for a little longer than the time that is already dedicated for that signal group in the planning of control decisions ($t^{\mathrm{gp}}_{\lambda,k}$). The idea behind introducing this variable is that the effects of wrong assumptions in the planning can be reduced (e.g. a vehicle accelerates slower than expected, the queue discharge rate appears to be lower than the one the planning is based on, there are more vehicles present than expected). By setting this variable to zero, the algorithm will not allow signal group λ to be green longer than the green time that is allocated to this signal group in the planning, which results in more accurate time-to-green and time-to-red predictions, but which is at the expense of reducing the delay at the same time. If a queue dissolves earlier than planned, e.g. because road users accelerate more aggressively than anticipated, then green simply ends on the basis of detection like in standard Dutch traffic signal control. If green is extended during the additional green time, then the distant loop detector is dropped, which means that extension of green only takes place via the extension loop detector.

Prolonging Green (MG)

During the prolonging green state the signals of a signal group show green as long as giving green does not prevent other signal groups with green requests to become green. For this purpose Algorithm 4.3 is applied. This algorithm basically states that green for signal group λ is to end if there is a conflicting signal group λ^* with a green request, except when there is at least one signal group $\lambda^{'}$ that prevents signal group λ^* from becoming green, while the internal state of signal group $\lambda^{'}$ equals either fixed green or extension green.

Algorithm 4.3: Algorithm for extending green during the prolonging green time period (MG)

Input : $\alpha_{\lambda,k}$, FC, $t^{\mathrm{cl}}_{\lambda^*,\lambda}$, $t^{\mathrm{fg}}_{\lambda}$, $t^{\mathrm{gd}}_{\lambda,k}$, $t^{\mathrm{mg}}_{\lambda}$, $t^{\mathrm{vg2}}_{\lambda}$
Output: $s^{\mathrm{status}}_{\lambda,k}$

foreach $\lambda \in FC$ **do**
  if $t^{\mathrm{gd}}_{\lambda,k} \leq (t^{\mathrm{fg}}_{\lambda} + t^{\mathrm{vg2}}_{\lambda} + t^{\mathrm{mg}}_{\lambda}) \wedge t^{\mathrm{mg}}_{\lambda} > 0$ **then**
    foreach $\lambda^* \in FC$ **do**
      if $t^{\mathrm{cl}}_{\lambda,\lambda^*} > -99 \wedge \alpha_{\lambda,k} = True$ **then**
        set $s^{\mathrm{status}}_{\lambda,k} = GL$
        foreach $\lambda^{'} \in FC$ **do**
          if $t^{\mathrm{cl}}_{\lambda^{'},\lambda^*} > -99 \wedge (s^{\mathrm{status}}_{\lambda^{'},k} = FG \vee s^{\mathrm{status}}_{\lambda^{'},k} = VG)$ **then**
            set $s^{\mathrm{status}}_{\lambda,k} = MG$
          end
        end
        if $s^{\mathrm{status}}_{\lambda,k} = GL$ **then**
          break
        end
      end
    end
  else
    propagate to GL by setting $s^{\mathrm{status}}_{\lambda,k} = GL$
  end
end

Amber (GL)

During the amber period the signal shows amber for the duration of the fixed amber time. The internal status propagates to red before request (RV) if the condition is met that is described by the following equation:

$$\text{if} \quad t^{\mathrm{yd}}_{\lambda,k} \geq t^{\mathrm{y}}_{\lambda} \quad \text{propagate to RV by setting} \quad s^{\mathrm{status}}_{\lambda,k} = RV \tag{4.15}$$

4.2.4. Updating Signal Durations and Timers

In the fourth and final step of the TSC algorithm the signal durations and a number of timers are updated. The next paragraphs explain the activities in this step in detail.

Updating Signal Durations

The colours of the signals for each signal group $\lambda \in FC$ have been set during the signal completion process and hence it is now time to update the signal duration timers. This is a rather straightforward step and it is performed according to the following equations:

$$
\begin{aligned}
t^{\mathrm{rd}}_{\lambda,k} &= \begin{cases} 0 & \text{if } s^{\mathrm{colour}}_{\lambda,k} = \mathit{amber} \vee \mathit{green} \\ t^{\mathrm{rd}}_{\lambda,k-1} + 0.1 & \text{if } s^{\mathrm{colour}}_{\lambda,k} = \mathit{red} \end{cases} && \forall \lambda \in FC \\
t^{\mathrm{yd}}_{\lambda,k} &= \begin{cases} 0 & \text{if } s^{\mathrm{colour}}_{\lambda,k} = \mathit{red} \vee \mathit{green} \\ t^{\mathrm{yd}}_{\lambda,k-1} + 0.1 & \text{if } s^{\mathrm{colour}}_{\lambda,k} = \mathit{amber} \end{cases} && \forall \lambda \in FC \qquad (4.16) \\
t^{\mathrm{gd}}_{\lambda,k} &= \begin{cases} 0 & \text{if } s^{\mathrm{colour}}_{\lambda,k} = \mathit{red} \vee \mathit{amber} \\ t^{\mathrm{gd}}_{\lambda,k-1} + 0.1 & \text{if } s^{\mathrm{colour}}_{\lambda,k} = \mathit{green} \end{cases} && \forall \lambda \in FC
\end{aligned}
$$

Updating Timers

The following timers of the TSC algorithm are updated:

- Increase the current time k with one decisecond via:

$$k \leftarrow k + 0.1 \qquad (4.17)$$

- For each signal group $\lambda \in FC$ decrease the time-to-green and time-to-green predictions with one decisecond via the following equations:

$$
\begin{aligned}
TTG^{\mathrm{time}}_{\lambda,k} &\leftarrow TTG^{\mathrm{time}}_{\lambda,k} - 0.1 && \forall \lambda \in FC \\
TTR^{\mathrm{time}}_{\lambda,k} &\leftarrow TTR^{\mathrm{time}}_{\lambda,k} - 0.1 && \forall \lambda \in FC
\end{aligned} \qquad (4.18)
$$

- For each signal group $\lambda \in FC$ decrease the remaining green time according to the planning of control decisions if the corresponding signal is green:

$$\text{if } s^{\mathrm{colour}}_{\lambda,k} = \mathit{green} \text{ then } t^{\mathrm{gp}}_{\lambda,k} \leftarrow t^{\mathrm{gp}}_{\lambda,k} - 0.1 \qquad \forall \lambda \in FC \qquad (4.19)$$

- For each signal group $\lambda \in FC$ increase the current waiting time with one decisecond if the corresponding signal is red and there is a green request:

$$\text{if } s^{\mathrm{colour}}_{\lambda,k} = \mathit{red} \wedge \alpha_{\lambda,k} = \mathit{True} \text{ then } t^{\mathrm{wa}}_{\lambda,k} \leftarrow t^{\mathrm{wa}}_{\lambda,k} - 0.1 \qquad \forall \lambda \in FC \qquad (4.20)$$

- Finally, decrease the remaining time until recalculation of the planning of control decisions takes place with one decisecond via the following equation:

$$t^{\mathrm{recalculate}}_{k} \leftarrow t^{\mathrm{recalculate}}_{k} - 0.1 \qquad (4.21)$$

4.3. Summary

The following findings summarise the fourth chapter of this thesis:

- The new traffic signal control algorithm that is developed in this study is split up in two parts for multiple reasons. The first part includes the PCD algorithm, which determines a planning of control decisions. The second part is the TSC algorithm, which uses the planning that is created by the PCD algorithm to grant the right on green to signal groups. The TSC algorithm basically determines the colour for each of the signals at the intersection for every time instant. The TSC algorithm strictly controls when the PCD algorithms recalculates the planning of control decisions;
- Like most other traffic signal control algorithms, the TSC algorithm runs every tenth of a second. The TSC algorithm also embraces the concept of signal completion and it uses loop detectors for determine whether or not to end green, however, it does not use a block structure to determine what signal groups are allowed to become green, but instead it uses the planning by PCD algorithm for this purpose;
- By providing and extending green on the basis of loop detectors, the TSC algorithm is able to handle inaccurate vehicle arrivals, wrongly assumed queue discharge rates and other errors in the planning. Compensating for these inaccuracies is beneficial from a delay point of view, however, this is at the expense of the accuracy of the time-to-green and time-to-red predictions. In order to control the degree to which deviating from the planning is accepted, a parameter is included that allows the user to specify the maximum number of seconds a signal group can be given green in addition to the time that is already reserved for that signal group in the planning of control decisions. The balance between adaptability/delay reduction and prediction of control decisions is hence incorporated in the TSC algorithm;
- By not strictly copying the planning of control decisions and forcing signals to become green accordingly, but by only granting the right on green instead, it is prevented that the traffic signal controller gives green when there are in fact no vehicles. This situation could occur in case of wrong vehicle arrivals, which certainly needs to be taken into account, especially in a non-simulation environment;
- User-adjustability of the TSC algorithm greatly benefits by the introduction of a large number of variables, which include among other things fixed green times, detector gap times, maximum waiting times and additional green times.

5
Algorithm Evaluation

Having developed and described the new traffic signal control algorithm (PCD + TSC), the next and final step of this research is to evaluate the performance of the new algorithm. Section 5.1 describes the evaluation set-up, which includes a description of the scenarios (Section 5.1.1) and a discussion of the reliability of the performance results (Section 5.1.2). The evaluation takes place in two manners, namely qualitative and quantitative, which Sections 5.2 and 5.3 elaborate on respectively. Section 5.4 summarises the chapter.

5.1. Evaluation Set-up

As explained in Chapter 1, this study's case study is the N65 intersection near the Dutch town of Helvoirt. Figure 5.1 shows that the intersection is located on the main route between the cities of Tilburg (+/- 200.000 inhabitants) and 's-Hertogenbosch (+/- 110.000 inhabitants), which results in high volumes of traffic passing this intersection on a daily basis. The N65 intersection is a clear example of a major-minor type of intersection and often congestion arises during peak hours[1].

Figure 5.1: The location of the N65 intersection near Helvoirt (red circle) with respect to its surroundings (figure source: Google, 2018).

[1] To limit congestion and reduce travel times, the Dutch government has very recently decided to invest large amounts of money into the N65 to introduce level-separation with on- and off-ramps, thereby discarding the traffic signal controllers (Omroep Brabant, 2018).

The performance of the new traffic signal control algorithm is evaluated using a microscopic traffic simulation model (Vissim). At the start of this study a calibrated Vissim network of the N65 intersection was already present, however, the existing network is modified for multiple reasons. The main modifications of the Vissim network are enumerated below and the modified network can be observed in Figure 5.2.

- The streaming areas that allow vehicles to queue in case of a red light are greatly increased for signal groups 05 and 06. Furthermore, the detection field has been standardised[2] and distant loop detectors have been added to allow for counting of vehicles;
- The streaming areas that allow vehicles to queue in case of a red light are somewhat increased for signal groups 10 and 11. Distant loop detectors have been added to allow for counting of vehicles;
- Parallel roads to the N65 have been removed on the north and south side of the intersection to smoothen discharging of queues at signal groups 05, 06, 10 and 11;
- Absolute priority is included for busses at signal groups 05 and 11;
- The split cyclist crossings on both sides of the intersection are merged to single cyclist crossings, and hence single signal groups, for simplicity.

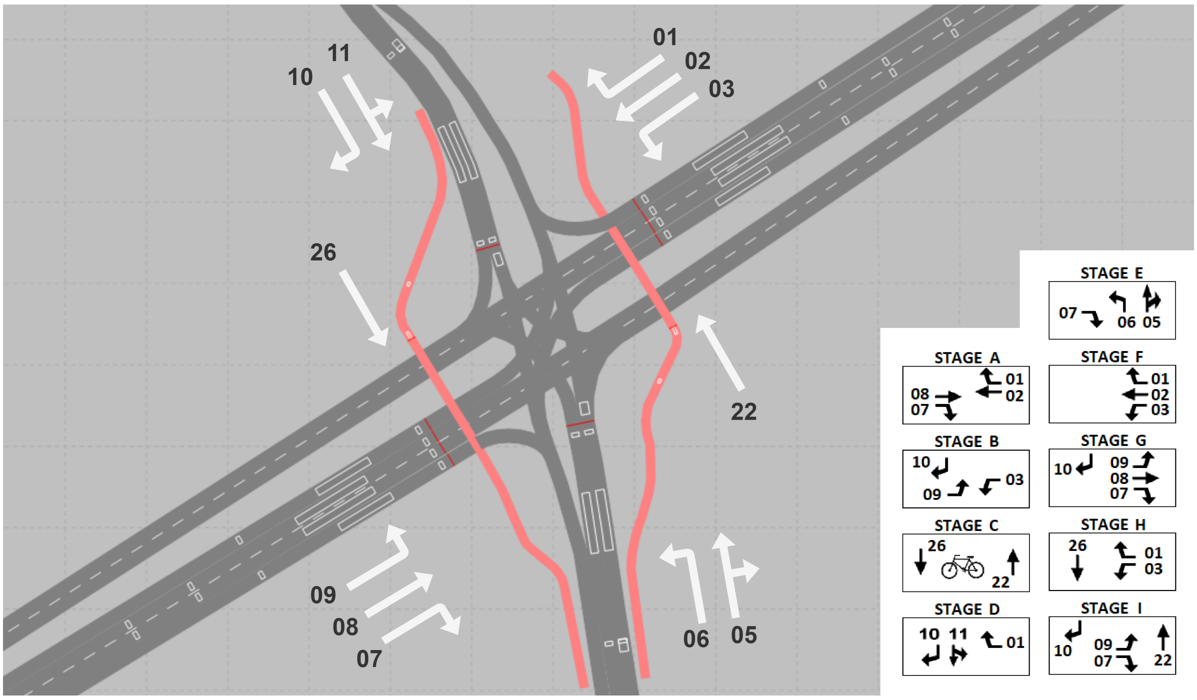


Figure 5.2: The modified Vissim network of the N65 intersection near Helvoirt. The white rectangles represent loop detectors and the red lines represent signal heads. The white arrows and black numbers indicate the present signal groups.

5.1.1. Scenarios

In simulating and evaluating the performance of the newly developed traffic signal control algorithm, a number of different scenarios are considered on the basis of three degrees of freedom. These degrees of freedom are (a) the type of traffic signal control algorithm, (b) the traffic flow and (c) the availability of vehicle arrivals. Due to the limited amount of time that is available in this study for evaluation, only two sets of traffic flows are considered (equal flows and major-minor) together with three different traffic signal control algorithms. For the newly developed traffic signal control algorithm the distinction between having vehicle arrivals and not having vehicle arrivals is also considered. In total this restricts the number of scenarios to eight (see Table 5.1). The next paragraphs elaborate on the details of the degrees of freedom.

[2] The length of the extension loop detectors has been increased from 1 meter to 20 meters and an additional extension loop detector has been removed. By equalising the detection field of signal groups 05 and 06 to the detection field of other signal groups, the need for separate green extension formulations for these signal groups is removed, which reduces the complexity of the programming code.

Number	*Traffic Signal Controller*	*Traffic Flows*	*Vehicle Arrivals*
1	Actuated	Equal flows	Not available
2	Actuated	Major-minor	Not available
3	Golden Controller	Equal flows	Not available
4	Golden Controller	Major-minor	Not available
5	PCD + TSC	Equal flows	Not available
6	PCD + TSC	Major-minor	Not available
7	PCD + TSC	Equal flows	Available
8	PCD + TSC	Major-minor	Available

Table 5.1: A list of scenarios and their characteristics.

Traffic Signal Controllers
The ’actuated’ traffic signal controller in the second column of Table 5.1 represents the original traffic signal control algorithm that was programmed by Goudappel Coffeng in 2010 and that has been updated multiple times over the past years, most recently in 2018. Like other traffic signal control algorithms of this type, the original algorithm considers a fixed block structure and set maximum green times. The second traffic signal control algorithm that is considered is Goudappel Coffeng’s Golden Controller. As explained in Chapter 1, this algorithm dynamically orders signal groups for their green time and maximum green times for signal groups are based on the number of vehicles in a queue plus the expected number of vehicle arrivals during green given the current traffic flow. The third and final algorithm is the algorithm that is newly developed in this study. Like the Golden Controller the new traffic signal control algorithm considers dynamic ordering of signal groups, however, this order is based on a planning of control decisions. Maximum green times for signal groups are obtained via the planning of control decisions too.

Traffic Flows
The equal flows scenario contains approximately equal flows for each of the signal groups taking into account the number of lanes. The exact traffic flows can be observed in the third column of Table 5.2. Scenarios with equal traffic flows form the ideal situation for the original actuated traffic signal controller, since it supports a fixed block structure in which signal groups are only included once in a cycle relatively well. In the major-minor scenarios traffic flows are increased by a factor 1.5 for signal groups 02 and 08, which better represents the actual traffic situation at the N65 intersection, and which is beneficial for both the Golden Controller and the newly developed traffic signal control algorithm. Due to higher traffic presence at the N65 in the latter scenarios, more frequent green realisations for signal groups 02 and 08 relative to other signal groups are expected, which should result in fewer overall delay. Exact traffic flows for the major-minor scenarios can be observed in the fourth column of Table 5.2.

Signal Group	*# Lanes*	*Equal Flows*	*Major-Minor*
01	1	200	200
02	2	600	900
03	1	200	200
05	1	200	200
06	1	200	200
07	1	200	200
08	2	600	900
09	1	200	200
10	1	200	200
11	1	200	200
22	1	50	50
26	1	50	50

Table 5.2: Traffic flows in vehicles/cyclists per hour for the ’equal flows’ and ’major-minor’ scenarios.

Vehicle Arrivals
In the first six scenarios vehicle arrivals are not available. In these cases vehicle arrivals are created manually on the basis of the traffic flow to prevent that too few green time is incorporated. In the final two scenarios vehicle arrivals are available and they are obtained via the Run and Repeat method that is explained in Section 1.5. It is expected that when vehicle arrivals are known delay reduction and time-to-green and time-to-red accuracy increase slightly.

5.1.2. Reliability of Performance Results

Since only a limited amount of time is available for the evaluation of the new traffic signal control algorithm, each of the scenarios in Table 5.1 is run only ten times. Via the following equation by Salomons (2015) a statement can be made with respect to the reliability of the performance results:

$$N^{'} \geq t^2_{\frac{1}{2}*\alpha,N-1} * (1 + \frac{1}{2} * \epsilon^2) * \frac{X_s^2}{X_d^2} \tag{5.1}$$

where α represents the reliability, $N^{'}$ is the number of simulation runs, X_s equals the sample standard deviation, X_d is the accepted standard deviation, ϵ is the normal distribution excess value and $t_{\frac{1}{2}*\alpha,N-1}$ equals the student t-distribution value.

In order to determine the values of some of the above mentioned variables, a starting point needs to be picked. Scenario 6 is considered as starting point[3] since the performance of this scenario lays closest to the average performance over all scenarios in which the new traffic signal control algorithm is evaluated.

For the total delay of scenario 6 a sample standard deviation X_s holds of approximately 5411 seconds (see Section 5.3.2). Since ten runs are performed, $N^{'}$ naturally equals 10. Also, given that the average is considered, the normal distribution excess value ϵ equals 0.

If an accepted deviation is considered that is equal to the sample standard deviation ($X_d = X_s = 5411$), then Equation 5.1 reduces to:

$$10 \geq t^2_{\frac{1}{2}*\alpha,10-1} \tag{5.2}$$

In this case $t_{\frac{1}{2}*\alpha,N-1}$ equals approximately 3.162, which means that a reliability of the performance results is achieved of almost 99%.

If an accepted deviation of for instance 3800 seconds is considered, then Equation 5.1 reduces to Equation 5.3. In this equation $t_{\frac{1}{2}*\alpha,N-1}$ equals approximately 2.246, which means that a reliability of the performance results is achieved of about 95%.

$$10 \geq t^2_{\frac{1}{2}*\alpha,10-1} * \frac{5411^2}{3800^2} \tag{5.3}$$

Although high reliability values are achieved (which indicates that the performance results that are presented in this chapter can be considered quite certain), it must be mentioned that the evaluation that was carried out is fairly limited. That is, only one intersection is considered and only eight different scenarios are executed. A wider comparison that includes additional intersections and a larger number of scenarios is suggested to allow for more certain performance results.

5.2. Qualitative Analysis

Visual observation of different simulation runs of various scenarios shows that the new traffic signal control algorithm is able to deal with the traffic well (e.g. no queues arise). The dynamic structure of the algorithm that allows signal groups to become green in any order is clearly visible and the choices that the traffic signal controller herein makes seem logic. If green is given to a signal group its queue is typically nicely cleared without giving away unnecessary green time, which implies that giving green on the basis of (a) the loop detectors, (b) the maximum green time according to the planning and (c) the maximum additional green time works well. The main notable points with respect to the performance of the traffic signal controller are:

[3] Choosing other scenarios as a starting point leads to different reliability levels. For scenarios with a larger sample standard deviation the reliability of the performance results drops if the accepted deviation is kept at a constant level. The inverse statement is also true.

- In the TSC algorithm alternative realisations are included in a fairly strict way. That is, a signal group λ may only realise alternatively if there is no conflicting signal group λ^* included in the planning that is planned to become green earlier than the sum of (a) the fixed green time of λ (b) the amber time of λ and (c) the clearance time between λ and λ^*. Even if there is a lack of, for instance, only one second, the signals from signal group λ will stay red. From the point of view of delay reduction relaxing the restrictions for alternative realisations might lead to gainings. Relaxing these restrictions is, however, at the expense of the accuracy of time-to-green and time-to-red predictions[4]. In a next version of the new traffic signal control algorithm the restrictions for alternative realisations could be made user-adjustable so that road authorities may decide about the strictness of the restrictions themselves;
- The check for signal groups to realise alternatively takes place in the order in which signal groups are included in the vector of signal groups *FC*. Due to the fact that the cyclist signal groups are located at the end of *FC*, often other signal groups are already allowed to realise alternatively before the cyclist signal groups may do so. Given the low volume of cyclists, which makes realising them primarily in the planning uninteresting, and given that the cyclist signal groups have a much stricter maximum waiting time constraint than all other signal groups[5], cyclists are regularly given green via a priority realisation on the basis of the maximum waiting time constraint. Such realisations almost always prevent the traffic signal controller from giving green to signal groups in the order in which delay reduction is minimal. Realising the cyclist signal groups earlier as alternative realisation instead of another signal group may in the end, however, have led to fewer overall delay, since a priority realisation for the cyclists could have been prevented. A solution for this that could be implemented in a next version of the new traffic signal control algorithm is to check signal groups for alternative realisations on the basis of the remaining time until their maximum waiting time is reached, so that forced realisations are prevented as much as possible. Other strategies for alternative realisations could also be thought of, such as realising signal groups with many conflicts over others;
- Priority realisations on the basis of exceeding the maximum waiting time constraint are included in the new algorithm on a stricter level than in both other traffic signal control algorithms. That is, green of a signal group is cut off after at least the fixed green time has elapsed if - at the time of recalculation of the optimal control policy - there is a conflicting signal group with a waiting time that exceeds the maximum waiting time. A cut off of green also happens if this leads to many vehicles remaining in the queue. In such cases both other traffic signal control algorithms allow for realisation of the full green time after which the signal group with priority is realised, which is much more convenient from the point of view of delay reduction. The problem of cutting of green is especially relevant in the following situations:
  - The amount of green time that is required for dissolvement of a queue is underestimated. In this case recalculation of the optimal control policy takes place at a time that queues have not cleared;
  - A short planning horizon is considered (e.g. 25 seconds) and traffic flows require/allow for long green times (e.g. 60 seconds). In this case recalculation of the optimal control policy also takes place at a time that queues have not cleared.
- A signal group λ that is currently in the prolonging green state (MG) turns amber and red if another signal group λ^* that covers λ also turns amber and red[6]. This also happens if λ, by chance, is in the next stage of the planning too. In this latter case the ideal situation is, however, described by λ staying green rather than turning amber and red after which green can reoccur, since turning amber and red might lead to unnecessary delay for vehicles as is explained in Figure 5.3.

  The situation that is explained in Figure 5.3 occurs if a signal group λ is in both the current and next stage, while there is too few traffic during the execution of the current stage to keep the signals of λ green on the basis of extension green (VG). This situation mostly, but not only, happens for right turning signal groups, since these signal groups can easily be included in multiple stages thanks to a limited number of conflicts, which results in much green time and short queues and which in turn leads to quick progressing from the extension green state (VG) to the prolonging green state (MG).

[4] Relaxing the restrictions for alternative realisations is especially useful if signal group λ^* is realised somewhat later than expected because of a conflicting signal group being green slightly longer than planned, since realising λ alternatively could then have happened without any consequences for λ^*.

[5] The maximum waiting time is set to 120 seconds for all signal groups, except for the cyclist signal groups which have a maximum waiting time of 60 seconds. The same values are considered in the Golden Controller.

[6] A signal group λ that is green in the prolonging green state (MG) can only remain green as long as there is at least one other signal group λ^* in the fixed green state (FG) or extension green state (VG) that prevents a third signal group λ' that conflicts λ from realising. Hence, if λ^* is the only signal group that covers λ and green of λ^* ends, then green of λ also ends.

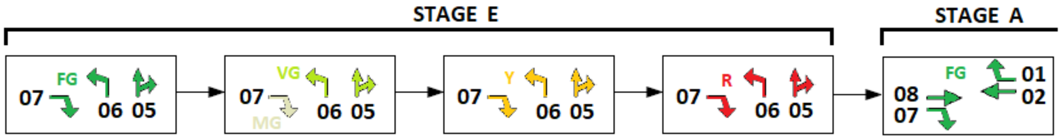


Figure 5.3: An example situation where green of a signal group is ended if that signal group is in two successive stages while there is too low traffic demand. Vehicles that arrive during the amber and red time of signal group 07 experience unnecessary delay, since the signals could technically have stayed green. In the figure time progresses from left to right. The abbreviations of FG, VG, MG, Y and R relate to fixed green, extension green, prolonging green, amber and red respectively.

Unwanted early ending of green also happens in case green of a signal group is realised earlier than planned, after which a large enough gap in the traffic flow arises to end green. To explain this, please consider the following example. Let signal group 05 be currently green and let signal group 02 be the following signal group to receive green to allow a platoon of vehicles that is expected to arrive at signal group 02 to pass the intersection. If signal group 05 finishes earlier than expected, then signal group 02 can become green earlier too, which might mean that the queue at signal group 02 has dissolved some time before the platoon of vehicles arrives, which in turn causes a gap between these flows of traffic and which might end green if the conditions for prolonging green do not hold. Since the duration of amber and guaranteed red is often longer than the time that vehicles require to cover the distance between the distant loop detector and the stop line (green is requested again via the distant loop detector), unnecessary delay for vehicles in the platoon arises. Unwanted ending of green could be solved by creating a more strict coupling with the planning, however, this is inconvenient in case of inaccurate vehicle arrivals;

- In case of a foreseen priority realisation (e.g. a public transport vehicle is expected to arrive at the intersection in several seconds) the TSC algorithm typically nicely clears the queue in front of the vehicle that is to be prioritised so that that particular vehicle is able to pass the intersection relatively freely. The time that signal groups may be green additional to the green time in the planning sometimes throws a spanner in the works, however, one could decide to disallow additional green time for conflicting signal groups in case of a priority realisation;
- Periods where the sequence of stages (CP^{stages}, see Chapter 3) is relatively stable, hence where the next stage equals the first stage after recalculation, alternate with more unstable periods. Part of this instability can be explained by the limited planning horizon that is considered during the simulation runs[7]. As the next example clarifies, the instability can also be explained by the working of the PCD algorithm itself, which is also the reason why the easy-to-understand sequence of stages (CP^{stages}) may present a distorted view of what is actually going on at the intersection.

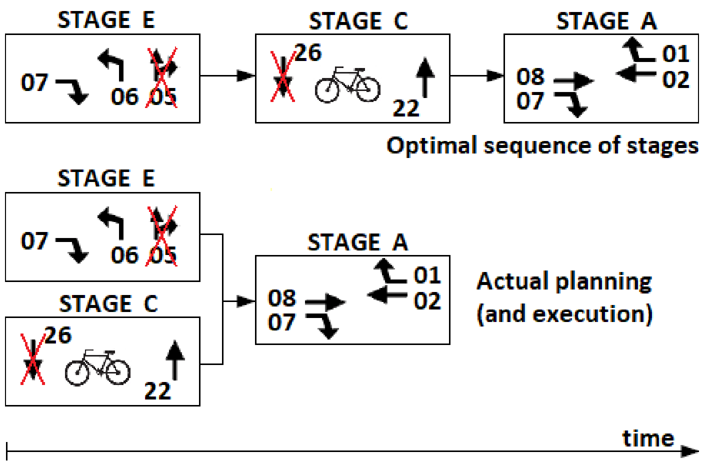


Figure 5.4: An example situation where the sequence of stages presents a distorted view of what is about to happen at the intersection.

[7] For computation time reasons a planning horizon of only 25 seconds is considered. From the point of view of delay reduction such a planning horizon should be sufficient, since Robertson and Bretherton (1974) mention that optimality in control is not particularly sensitive to vehicle arrivals beyond 25 seconds in the future.

Let there be green requests for signal groups 01, 02, 06, 07, 08 and 22 and let's assume that giving green to signal groups 06, 07 and 22 first and to signal groups 01, 02, 07 and 08 second minimises the delay. Signal groups 06, 07 and 22 are not represented by a single stage and hence two stages are required for this in the planning. Signal groups 06 and 07 receive green via stage E and signal group 22 receives green via e.g. stage C (see Figure 5.4). Although signal groups 06, 07 and 22 can and will be given green simultaneously due to the absence of traffic at signal groups 05 and 26, the PCD algorithm can only take this parallel realisation into account by planning stages E and C in successive order. This means that although stages E and C are displayed after each other in the optimal sequence of stages, these stages are in fact executed at the same time, making stage A the actual second stage in time. This principle has no effect on time-to-green predictions since the PCD algorithm takes ahead-of-time realisations into account (as explained in Sections 3.1 and 3.3, this is the basic idea of a signal-group based approach);

- The new traffic signal control algorithm seems to structurally slightly underestimate the time that is required to clear a queue. The new algorithm namely often recalculates on the basis of $t^{\text{recalculate}}$ at a time that there are still some vehicles remaining in a queue. This especially holds if a lorry is part of a standstill queue, which like any other vehicle is counted as a single vehicle, but which accelerates much more slowly. Although the algorithm may decide to lengthen green for signal groups that have vehicles remaining in queues by planning the current stage as the first stage in a new planning, the algorithm does not always make this choice, which, despite the additional green time, sometimes results in partial/incomplete queue dissolvement (one or two vehicles remain in a queue). By adjusting the user-adjustable queue discharge rates that the PCD algorithm uses, thereby deviating from the values for the queue discharge rates that the Golden Controller considers, this problem should be solved;
- In rare situations the TSC algorithm forces the PCD algorithm to recalculate every tenth of a second for the duration of a few seconds. This is unwanted behaviour and it indicates that there is at least one (currently unknown) combination of events in which the PCD algorithm recalculates that has not been thought of during the development of the TSC algorithm. Although in a simulation environment unnecessary recalculations have no impact (a simulation can easily be paused), they do cause problems in a real-life application of the algorithm and solving them is considered future work.

5.3. Quantitative Analysis

The second part of the evaluation considers a quantitative analysis. For this analysis performance indicators are defined in Section 5.3.1 and the performance results are elaborated on in Section 5.3.2.

5.3.1. Performance Indicators

The following performance indicators are defined for this study:

- The *delay* denotes the time that a vehicle requires to cover the distance between location A and location B minus the time that that vehicle requires to cover this distance in free-flow traffic conditions;
- The *queue length* denotes the absolute length of a queue;
- The *time-to-green* and *time-to-red accuracy* denotes the deviation between the predicted time-to-green/time-to-red time and the actual time-to-green/time-to-red time and is calculated likewise. This performance indicator only applies for the new algorithm, since both other algorithms do not present time-to-green and time-to-red predictions.

Each simulation run that is performed in this study considers a duration of 5400 seconds or 1.5 hours. During these runs the following data is obtained:

- Delays for each direction (e.g. north to south, north to east) over a simulation horizon of 1 hour and 15 minutes ($t = 900$ to $t = 5400$) aggregated over time intervals of 5 minutes;
- Queue lengths for each signal group $\lambda \in FC$ over a simulation horizon of 1 hour and 15 minutes ($t = 900$ to $t = 5400$) aggregated over time intervals of 1 minute;
- Time-to-green and time-to-red accuracies for each signal group $\lambda \in FC$ over the full simulation horizon of 1.5 hours ($t = 0$ to $t = 5400$).

5.3.2. Performance Results

The next paragraphs describe the performance of the new traffic signal control algorithm in terms of the performance indicators of Section 5.1.

Delay
In Tables 5.3 and 5.4 the total delay in seconds for each of the simulation runs can be observed for respectively the scenarios with equal traffic flows and major-minor traffic flows. As becomes clear from the tables, the traffic signal control algorithm that is developed in this study outperforms the original actuated algorithm by quite a bit, especially when vehicle arrivals are available. The new algorithm is, however, not yet able to achieve the same delay reduction that the Golden Controller achieves. Further perfection, which should be based on the points of improvement that are named throughout Section 5.2, is required to further increase delay reduction of the new traffic signal control algorithm.

As expected, the new algorithm performs better when vehicle arrivals are known. Though, due to the nature of vehicle generation in Vissim, the increase of the performance effect is only small. That is, the default distribution that is used in Vissim to generate traffic typically does not create large platoons of traffic, while especially platoons[8] are beneficial for the new traffic signal control algorithm. Also, in line with the expectation, the new algorithm performs slightly better in case of unequal traffic flows.

Random Seed	Actuated	Golden Controller		PCD + TSC		PCD + TSC (vehicle arrivals)	
1	101989	92350	90,5%	99742	97,8%	96695	94,8%
2	102160	94128	92,1%	100768	98,6%	95206	93,2%
3	101881	98771	96,9%	99147	97,3%	99946	98,1%
4	102560	93946	91,6%	95766	93,4%	93800	91,5%
5	103874	97753	94,1%	101323	97,5%	100926	97,2%
6	101441	94957	93,6%	103822	102,3%	98750	97,3%
7	103937	88711	85,4%	98739	95,0%	99206	95,4%
8	105004	93969	89,5%	100448	95,7%	99316	94,6%
9	103887	98032	94,4%	99480	95,8%	98748	95,1%
10	102479	88711	86,6%	98258	95,9%	99178	96,8%
Average:	102921	94133	91,5%	99749	96,9%	98177	95,4%

Table 5.3: Total delay in seconds for the equal flows scenarios. The percentages indicate the share of the performance with respect to the performance of the original actuated traffic signal controller.

Random Seed	Actuated	Golden Controller		PCD + TSC		PCD + TSC (vehicle arrivals)	
1	147705	134034	90,7%	139903	94,7%	140234	94,9%
2	143251	132095	92,2%	146419	102,2%	138529	96,7%
3	153322	136146	88,8%	146146	95,3%	149665	97,6%
4	142184	128542	90,4%	130408	91,7%	129323	91,0%
5	148270	136887	92,3%	148060	99,9%	148025	99,8%
6	150238	145341	96,7%	147294	98,0%	140037	93,2%
7	149030	127936	85,8%	141258	94,8%	139527	93,6%
8	151364	126934	83,9%	139122	91,9%	143816	95,0%
9	145695	140664	96,5%	143640	98,6%	145082	99,6%
10	150529	138398	91,9%	146378	97,2%	138201	91,8%
Average:	148159	134698	90,9%	142863	96,4%	141244	95,3%

Table 5.4: Total delay in seconds for the major-minor scenarios. The percentages indicate the share of the performance with respect to the performance of the original actuated traffic signal controller.

The average delay per vehicle on the level of signal groups can be observed in Tables 5.5 and 5.6.

[8] Thanks to traffic signal controllers upstream of the case study intersection, platooning regularly occurs for the south-eastern and north-western sides of the intersection (i.e. signal groups 01/02/03 and 07/08/09).

Direction	Signal Group	Traffic flow (weight factor)	Actuated	Golden Controller		PCD + TSC		PCD + TSC (vehicle arrivals)	
1	01	200	10,72	13,05	121,7%	11,23	104,8%	10,73	100,1%
2	02	900	32,46	24,51	75,5%	22,63	69,7%	22,49	69,3%
3	03	200	33,62	31,82	94,6%	35,04	104,2%	35,30	105,0%
4	05	100	34,80	35,95	103,3%	41,22	118,4%	40,73	117,0%
5		100	33,66	35,32	104,9%	39,78	118,2%	40,89	121,5%
6	06	200	33,78	34,92	103,4%	41,25	122,1%	41,85	123,9%
7	07	200	10,35	10,91	105,4%	10,02	96,8%	9,21	89,0%
8	08	900	31,66	23,75	75,0%	20,83	65,8%	20,04	63,3%
9	09	200	27,06	28,91	106,8%	30,80	113,8%	30,76	113,7%
10	10	200	24,00	17,51	73,0%	16,90	70,4%	16,29	67,9%
11	11	100	33,66	38,03	113,0%	58,91	175,0%	57,30	170,2%
12		100	34,25	38,91	113,6%	60,56	176,8%	58,65	171,2%
22	22	50	31,00	50,61	163,3%	50,72	163,6%	48,07	155,1%
26	26	50	25,10	42,14	167,9%	52,94	210,9%	51,20	204,0%

Table 5.5: Average delay per vehicle in seconds for the equal flows scenarios. The percentages indicate the share of the performance with respect to the performance of the original actuated traffic signal controller.

Direction	Signal Group	Traffic flow (weight factor)	Actuated	Golden Controller		PCD + TSC		PCD + TSC (vehicle arrivals)	
1	01	200	10,23	12,83	125,4%	11,40	111,4%	11,12	108,7%
2	02	900	35,03	27,07	77,3%	25,33	72,3%	23,78	67,9%
3	03	200	41,45	41,95	101,2%	43,68	105,4%	43,34	104,6%
4	05	100	43,01	47,14	109,6%	56,54	131,5%	58,02	134,9%
5		100	40,03	43,98	109,9%	57,14	142,7%	56,40	140,9%
6	06	200	42,53	44,49	104,6%	55,04	129,4%	54,68	128,6%
7	07	200	10,11	10,71	105,9%	9,94	98,3%	9,92	98,1%
8	08	900	34,86	26,87	77,1%	23,74	68,1%	22,52	64,6%
9	09	200	35,10	36,39	103,7%	36,53	104,1%	40,54	115,5%
10	10	200	32,48	21,77	67,0%	22,74	70,0%	23,00	70,8%
11	11	100	40,91	50,36	123,1%	80,53	196,8%	79,29	193,8%
12		100	43,08	51,81	120,3%	80,93	187,9%	83,79	194,5%
22	22	50	38,00	57,58	151,5%	55,09	145,0%	53,23	140,1%
26	26	50	30,17	47,05	155,9%	55,02	182,4%	58,06	192,4%

Table 5.6: Average delay per vehicle in seconds for the major-minor scenarios. The percentages indicate the share of the performance with respect to the performance of the original actuated traffic signal controller.

The following stands out from Tables 5.5 and 5.6:

- One of the main differences between the new algorithm and both other algorithms is that the new algorithm chooses to give more green to signal groups 02 and 08, of which signal groups 01 and 07 also benefit. The new algorithm thereby increases the level of inequality between individual signal groups. By doing so, though, it achieves a further reduction of the delay for vehicles at signal groups 02 and 08 (most vehicles pass the intersection here), at the expense of an increase of the delay for vehicles at most other signal groups;
- In both the Golden Controller and the new algorithm cyclists see their waiting time increase quite drastically. This is a direct result of these signal groups being given green less frequently, which has to do with (a) the low volume of cyclists at the intersection and (b) the lack of priority measures for these road users (i.e. in this evaluation all signal groups are considered equal). Compared to the Golden Controller the new algorithm achieves a slight delay reduction for signal group 22, however, the delay increases for signal group 26. On average, the Golden Controller leads to slightly lower waiting times for cyclists;
- Another remarkable point considers the large increase of delay for vehicles at signal group 11. This has to do with green realisations of signal group 22 and is explained as follows.

  A primary realisation of signal group 22 can only take place via two stages, namely $[07, 09, 10, 22]$ and $[22, 26]$. Due to the low volume of cyclists, almost always the first stage is chosen. This means that green is given to signal groups 07, 09, 10 and 22, which leads to dissolving of queues here, which in turn reduces the potential for a primary realisation of stages that include these signal groups. Since signal group 11 is included in only one stage, namely [01, 10, 11], and since signal group 01 already receives much green time, realisation of this latter stage pretty much fully relies on signal group 11. Since traffic flows are high for signal groups that conflict signal group 11, especially in the major-minor scenarios, the algorithm often chooses to grant the right on green to the signal groups of other stages (e.g. [01, 02, 07, 08]) over stage [01, 10, 11], which leads to large delays[9] at signal group 11.

Adding a stage to the set of stages *SG* that consists of signal groups 01, 11 and 26 ([01, 11, 26]) could be attractive for the performance of the algorithm. That is, if signal group 26 is to be realised with priority, which due to a strict maximum waiting time constraint is the case most of the time when signal group 26 is realised, then the possibility is added for signal group 11 to become green too. The same principle holds for signal group 05 in combination with signal group 26. To prevent long waiting times, one could also decide to increase the weight factor for signal group 11 so that realising signal group 11 becomes more attractive in the PCD algorithm. One could do the same for the cyclist signal groups.

Queue Length

Tables 5.7 and 5.8 present the 95th percentile of the queue length in meters for respectively the scenarios with equal traffic flows and major-minor traffic flows. The values in these tables show that the longest queues typically arise for signal group 11 and that queue lengths for signal groups 02 and - especially - 08 decrease, together with the queue length for signal group 10. The queue lengths in these tables thereby support the remarks that have been made in the previous paragraphs.

Signal Group	Actuated	Golden Controller		PCD + TSC		PCD + TSC (vehicle arrivals)	
01	25	30	120,0%	25	100,0%	25	100,0%
02	60	45	75,0%	45	75,0%	44	73,3%
03	48	43	89,6%	45	93,8%	48	100,0%
05	49	50	102,0%	53	108,2%	56	114,3%
06	45	46	102,2%	52	115,6%	51	113,3%
07	26	25	96,2%	25	96,2%	25	96,2%
08	57	49	86,0%	44	77,2%	44	77,2%
09	43	44	102,3%	44	102,3%	44	102,3%
10	43	33	76,7%	32	74,4%	32	74,4%
11	48	52	108,3%	72	150,0%	72	150,0%

Table 5.7: 95th Percentile of the queue length in meters for the equal flows scenarios. The percentages indicate the share of the performance with respect to the performance of the original actuated traffic signal controller.

Signal Group	Actuated	Golden Controller		PCD + TSC		PCD + TSC (vehicle arrivals)	
01	25	28	112,0%	25	100,0%	25	100,0%
02	93	75	80,6%	70	75,3%	70	75,3%
03	57	56	98,2%	57	100,0%	56	98,2%
05	57	62	108,8%	71	124,6%	70	122,8%
06	56	58	103,6%	68	121,4%	67	119,6%
07	26	26	100,0%	26	100,0%	28	107,7%
08	95	80	84,2%	73	76,8%	68	71,6%
09	52	52	100,0%	51	98,1%	55	105,8%
10	50	39	78,0%	38	76,0%	39	78,0%
11	55	68	123,6%	100	181,8%	100	181,8%

Table 5.8: 95th Percentile of the queue length in meters for the major-minor scenarios. The percentages indicate the share of the performance with respect to the performance of the original actuated traffic signal controller.

[9] The maximum waiting time constraint that is included for all signal groups $\lambda \in FC$, hence also for signal group 11, prevents too long waiting times.

Time-to-Green and Time-to-Red Accuracy

One of the components of the new algorithm is the ability to present time-to-green and time-to-red predictions to road users. Figures 5.5, 5.6, 5.7 and 5.8 include boxplots of the time-to-green and time-to-red accuracies for all signal groups $\lambda \in FC$ of scenarios 5, 6, 7 and 8 respectively. In these figures the lower and upper boundaries are represented by the 5^{th} and 95^{th} percentile. The figures in Appendix E present the time-to-green and time-to-red accuracy distributions.

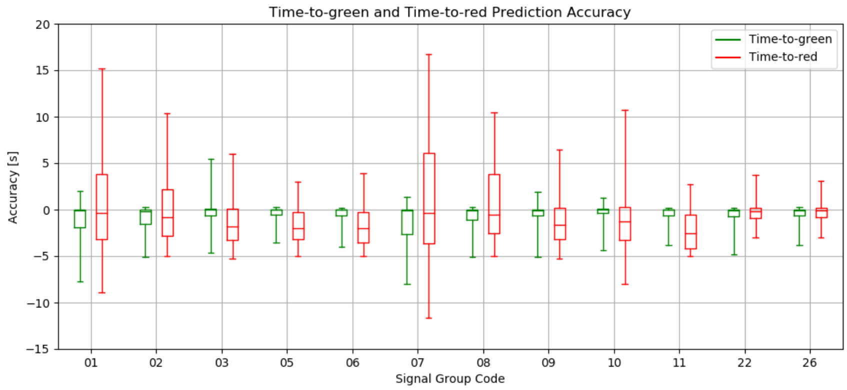


Figure 5.5: Time-to-green and time-to-red accuracies in seconds for all signal groups $\lambda \in FC$ for scenario 5 (equal flows, no vehicle arrivals available).

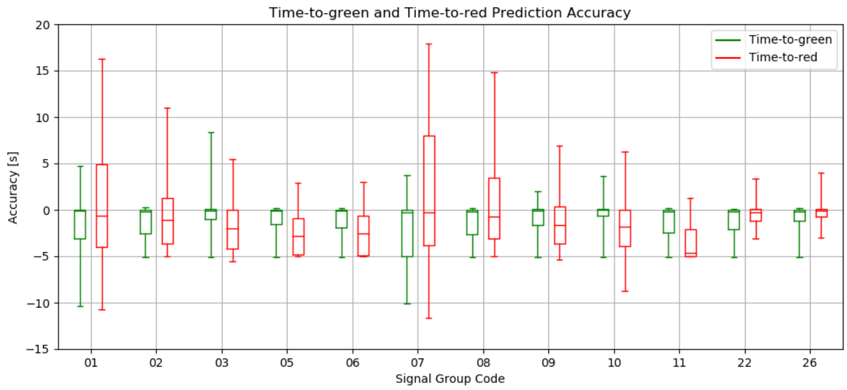


Figure 5.6: Time-to-green and time-to-red accuracies in seconds for all signal groups $\lambda \in FC$ for scenario 6 (major-minor, no vehicle arrivals available).

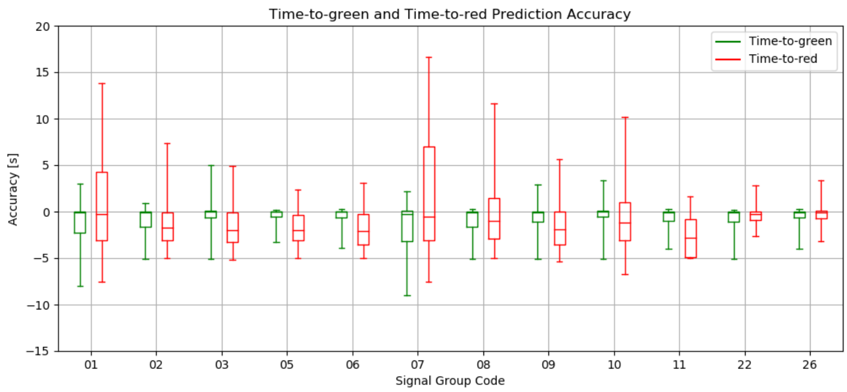


Figure 5.7: Time-to-green and time-to-red accuracies in seconds for all signal groups $\lambda \in FC$ for scenario 7 (equal flows, vehicle arrivals available).

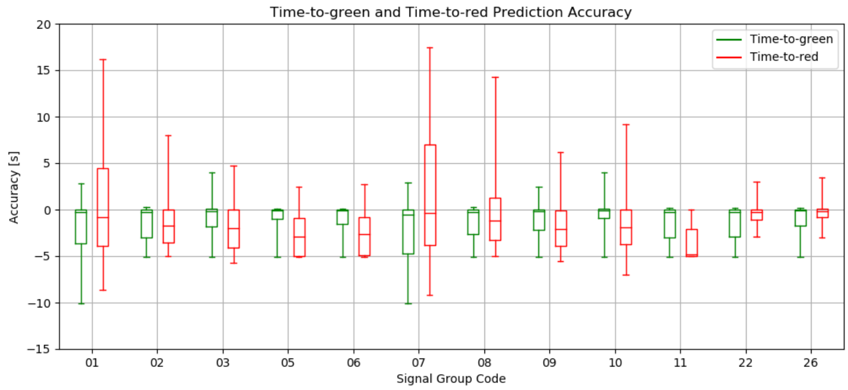


Figure 5.8: Time-to-green and time-to-red accuracies in seconds for all signal groups $\lambda \in FC$ for scenario 8 (major-minor, vehicle arrivals available).

Please note that during the simulation runs time-to-green and time-to-red accuracies are only determined for signal groups with certainty indication '=', which implies the following:

- The time-to-green and time-to-red accuracies that are presented in this section include primary and primary ahead realisations, prolonging green (MG) and extended green via the additional green time;
- The time-to-green and time-to-red accuracies that are presented in this section exclude alternative realisations. Also, time-to-green and time-to-red penalties for signal groups that were granted the right on green in a specific planning (the certainty indication for this signal group equals '='), but that have been left out in the successive planning (certainty indication '<>') are not considered. Penalties for signal groups that were planned in a planning (certainty indication '=') and that are postponed, yet still included, in a successive planning are included (here the certainty indication remains '=').

  The choice to exclude signal groups with certainty indication '<>' is made because the time-to-green and time-to-red accuracy for these signal groups is strongly dependent on the time-to-green and time-to-red values that are chosen for signal groups that are not present in the planning, which gives a poor view of the actual deviation. For plannings that include all of the signal groups, the subject of excluding signal groups plays no role (the certainty indication for each signal group is then namely '=').

Looking at Figures 5.5, 5.6, 5.7 and 5.8 and at the figures in Appendix E, the following stands out:

- The time-to-green and time-to-red predictions seem to be presenting the actual moments at which signals turn green and red quite well. For most signal groups the majority of accuracies is slightly negative, this holds for both time-to-green and time-to-red predictions, which indicates that green is given longer than planned, and which is in accordance with the observation in Section 5.2 that the required green time to dissolve a queue is underestimated;
- The time-to-red predictions for two of three right turning signal groups (i.e. 01 and 07) are clearly worse than for most other signal groups. The increased number of negative time-to-red values can be explained by extension of green via the prolonging green period (MG), which is not incorporated in the planning. Explaining the increased number of positive time-to-red values is more difficult. The principle that is explained in Section 5.2 in which the signals of a signal group λ that is in prolonging green (MG) turn amber and red if the signals of a signal group λ^* that covers λ also turn amber and red certainly plays a role here, however, this does not occur with the frequency that is suggested by the figures in Appendix E. Furthermore, from visual observations of the simulation runs prolonging green (MG) taking over green extension from extension green (VG) in case of a lack of traffic works well, hence this is also ruled out as a contributing factor to the increased number of positive time-to-red values.

5.4. Summary

This chapter's main findings are enumerated below:

- The new traffic signal control algorithm that is developed in this study is evaluated using microsimulation. The N65 intersection near Helvoirt, which is located on the main road between the Dutch cities of Tilburg and 's-Hertogenbosch and which is a perfect example of a major-minor type of intersection, plays the role of case study. The new algorithm is compared to two other traffic signal control algorithms, which are the original actuated algorithm (developed by Goudappel Coffeng in 2010 and most recently updated in 2018) and the Golden Controller (developed by Goudappel Coffeng in 2018). Although the performance results that this study provides have a fairly high certainty level (depending on the accepted deviation 95% to 99%), a wider comparison that includes different intersections and additional scenarios is suggested to increase the representativity of the conclusions drawn in this thesis;
- Visual observation of the simulation runs show that the algorithm runs fairly smoothly and that it is able to handle traffic well. This observation is confirmed in a quantitative comparison of the delay for vehicles at the intersection. That is, by reducing the total delay with almost 5%, the new algorithm clearly outperforms the original actuated traffic signal control algorithm. Due to various imperfections the new algorithm is not yet able to achieve the delay reduction of 9% that the Golden Controller achieves. The main imperfections of the new algorithm are:
  - Ordering of signal groups for alternative realisations is inefficient. That is, signal groups are currently checked for becoming green via an alternative realisation in the order in which signal groups are included in the set of signal groups *FC*, however, basing the order on the waiting time of signal groups or on the remaining time before the maximum waiting time is reached may be more beneficial for the performance of the algorithm;
  - The signals of a signal group sometimes turn amber and red and become green again directly thereafter if that stage is in both the current and next stage of the planning of control decisions. This problem occurs when there is too few traffic during the execution of the first stage to keep the signal green on the basis of extension green (VG), which mostly happens for right turning signal groups. Recurrence of green sometimes leads to unnecessary delay for vehicles, which is undesired;
  - Green is terminated in case of an (unplanned) priority realisation, even though there might be a queue of vehicles remaining. This problem is especially relevant when short planning horizons are considered or when the amount of green time for queue dissolvement is underestimated. This latter appears to be the case for the PCD algorithm.
- In handling traffic, the new traffic signal control algorithm makes somewhat similar decisions as the Golden Controller. That is, green time for signal groups 02 and 08 increases, while green time for most other signal groups decreases. This especially holds for cyclists, who see their waiting time increase quite drastically. The new algorithm seems to make more extreme control decisions than the Golden Controller, since it gives even more green time to signal groups 02 and 08 (which have a high traffic flows) and further reduce green time for most other signal groups. It is unclear whether increased green time is due to a higher frequency of green realisations, increased duration of green or a combination of both;
- With respect to the accuracy of time-to-green and time-to-red predictions, the average inaccuracy is a slightly negative value, which indicates that the amount of green time that is required for dissolvement of queues is underestimated. This can be resolved by changing the user-adjustable queue discharge rates. Time-to-green and time-to-red predictions are best for signal groups with many conflicts (since this often prevents prolonging green), and they are most worse for right turning signal groups (which typically have only few conflicting signal groups).

6

Conclusions and Future Work

In this final chapter the work that has been done is concluded. Section 6.1 provides the reader with the main findings of this study by answering the research questions of Chapter 1. Section 6.2 elaborates on future work, which includes improvement possibilities and a description of the actions that need to be taken before the algorithm can be applied in practice.

6.1. Conclusions

The research that is described in this thesis aims at developing a new traffic signal control algorithm that relies on predicted vehicle arrivals to make a short-term planning of control decisions. In developing this new algorithm, five research questions are defined. This section presents the answers to these questions, which gives insight into the conclusions of the research.

What is the definition of a vehicle arrival, which data is required from vehicles, and how can vehicle arrivals be determined from this data?

In order to achieve a reduction of the delay, the traffic signal controller is required to know when a vehicle really needs a green signal to pass the intersection. This means that a vehicle arrival for a signal group is defined to be the moment at which a vehicle is expected to arrive at the stop line of that particular signal group.

In this study vehicle arrivals are determined via the Run and Repeat approach. In this approach a simulation run is executed once during which locational data of vehicles is obtained, after which the exact same simulation run is executed again (same random seed), but now with the new traffic signal control algorithm being active (see Section 1.5). The locational data, which consists of (a) the distance between the location of a vehicle and the stop line the vehicle approaches and (b) the velocity of that vehicle, is then translated into vehicle arrivals.

Obviously, the Run and Repeat approach is only applicable in simulation environments. In practice different approaches to obtain vehicle arrivals need to be considered, such as translating detector occupancies in vehicle presence and extrapolating the location of these vehicles via a car-following model.

Due to the limited availability of vehicle arrivals in practice, vehicle arrivals are only considered additional data. By manually creating vehicle arrivals on the basis of traffic flows, the new traffic signal control algorithm is also able to handle the situation in which no vehicle arrivals are available at all. Nonetheless, the availability of vehicle arrivals leads to increased performance of the new traffic signal control algorithm.

How is a short-term planning of control decisions created and are there any existing algorithms that can, whether or not after modification, be reused for this purpose?

In the literature review part of this study (Chapter 2) various look-ahead traffic signal control algorithms are evaluated for their design choices and their potential to be reused. Although many of the algorithms present useful features such as a variable interval size of the planning horizon, it turns out that (direct) reuse of most of these algorithms is inconvenient. Arguments for this range from infeasible computation times (e.g. Goodall (2013)) to the application of approaches that do not support standard Dutch actuated/adaptive traffic signal control (e.g. NEMA, stage-based approach).

The existing algorithms that present the most promising approaches are the ones by Sen and Head (1997), van Katwijk (2008) and Chen and Sun (2016). These three algorithms try to find the sequence and duration of blocks/stages that fit a certain control objective best. The signal-group based approach that van Katwijk (2008) introduces is especially useful, since this approach incorporates ahead-of-time green realisations of signal groups, which perfectly fits the Dutch practice of traffic signal control. The new traffic signal control algorithms combines the best parts of the above mentioned three algorithms and it adds a number of new functions which includes, among other things, calculation of time-to-green and time-to-red predictions.

How can the objectives of predictability and priority be made user-adjustable and how can they, together with adaptability, be balanced within the new traffic signal control algorithm?

The Planning of Control Decisions (PCD) algorithm, which is part one of the new traffic signal control algorithm, uses optimisation to determine a sequence of control decisions of over a given planning horizon length with the purpose of minimising delays. The output of the algorithm consists of (a) the order and duration in which stages are granted the right on green and (b) green start and end times for all signal groups. The planning of control decisions that is generated by the PCD algorithm may represent the overall optimal solution, but this cannot be guaranteed. The PCD algorithm, of which the main inputs are the queues at the moment of recalculation and the expected vehicle arrivals, includes the balance between adaptability (i.e. delay reduction) and priority via user-adjustable weight factors. The algorithm is set up in a way that road authorities may prioritise (a) individual road users, (b) vehicle types and (c) specific signal groups. The weight factors influence the delay that vehicles are set to experience over a single time interval in the planning.

The Traffic Signal Control (TSC) algorithm, which is the second part of the new traffic signal control algorithm, uses the planning of control decisions by the PCD algorithm as a framework to grant the right on green to signal groups. The TSC algorithms basically determines the colour of all signals at every time instant and the balance between adaptability and predictability is included herein. That is, green of signal groups is extended on the basis of loop detectors rather than for the precise duration of the planning, which prevents unnecessary green time (and delay accordingly) from occurring when queues dissolve quicker than expected. At the same time, the TSC algorithm includes a parameter that allows signal groups to be green for some time additional to the green time that is considered in the planning, so that underestimation of traffic and other errors in the planning (e.g. queues dissolve slower than expected) can be dealt with.

How is computational efficiency of the new algorithm defined and achieved?

Computation time is a fundamental limitation in developing traffic signal control algorithms. Computational efficiency, which is defined as the degree to which the full set of control options has to be explored to find a control policy, is hence a key topic. This especially holds for look-ahead traffic signal control in which the set of control options is exposed to exponential growth.

The optimisation technique on which the PCD algorithm relies is a forward recursion dynamic programming formulation. This technique is introduced in traffic signal control by Sen and Head (1997) and is also applied by van Katwijk (2008) and Chen and Sun (2016). Dynamic programming, which reduces the time complexity from an exponential level to a polynomial level, allows for excluding a large set of control options to be explored. In essence, the PCD algorithm breaks the full horizon over which a planning is to be determined up in smaller, more manageable parts, after which the best control policies for these parts are combined into a single control policy for the full horizon.

Despite the above mentioned strategy, computation times of the PCD algorithm are still fairly high and the algorithm certainly is not able to run in real time. Planning horizons above approximately 90 seconds (in case of 4 stages) and 70 seconds (in case of 9 stages) are even unfeasible due to the computation time outgrowing the planning horizon.

How does the new algorithm perform in comparison to other traffic signal control algorithms?

Using microsimulation on a N65 case study intersection the performance of the new traffic signal control algorithm is compared to the performance of two other algorithms. These algorithms are (a) the original actuated traffic signal control algorithm and (b) the Golden Controller.

In terms of delay, the new traffic signal control algorithm outperforms the original algorithm by quite a bit. Depending on whether or not vehicle arrivals are available, the share of delay of the new algorithm is only 95.3%. The new algorithm is not yet able to achieve the same delay reduction that the Golden Controller achieves (share of delay: 91.6 to 90.9%). Various inefficiencies, which include unwanted and unnecessary early ending and recurring of green and inefficient alternative realisations, need to be tackled to ensure a further reduction of the delay.

Time-to-green and time-to-red predictions, which cannot be compared to both other algorithms, appear to be fairly accurate. That is, the majority of the (in)accuracy values lay close to zero, which means that the predicted time-to-green and time-to-red times largely equal the actual time-to-green and time-to-red times. On average the accuracy is slightly negative, which indicates that the time that is required for queues to dissolve is underestimated by the PCD algorithm. Time-to-green and time-to-red predictions for signal groups with many conflicts tend to be most accurate.

Based on how vehicle arrivals are taken into account, platooning is beneficial for the performance of the new algorithm. Via vehicle arrivals virtual coupling of intersections is also created. This is especially useful for intersections that are within a range of 300-500 meters from each other, for which it is often unclear if strict coupling is necessary. Furthermore, the new algorithm should not be applied to intersections with strongly fluctuating traffic flows while vehicle arrivals are unknown, since adding vehicle arrivals manually on the basis of average traffic flows then leads to large under-/overestimation of vehicle presence. This is less of a problem for the Golden Controller since this algorithm solely considers queues to determine the order in which signal groups become green, while the new algorithm also incorporates expected vehicle arrivals.

6.2. Future Work

The PCD and TSC algorithms that have been described in Chapters 4 and 5 are certainly not perfect and many more hours could be spent into extending the algorithms' functionalities and further refining their performance. By doing so, computation times can be reduced and an additional reduction of the delay can be achieved. Apart from discussing improvement possibilities for the algorithms, the next paragraphs also elaborate on research enhancements that could lead to e.g. a better motivation for the values of parameters and more reliable performance results. The next paragraphs are set up on the basis of this thesis' chapters, so that future work is discussed over the full width of the study.

Literature Review

Exploration of search techniques is limited to the field of traffic engineering in this study. There might, however, be more efficient algorithms in non-traffic related work fields such as bioinformatics, operations research and computer science (Bhowmik, 2010) that could be modified to determine a planning of control decisions with it. Also, more attention could be given to state-of-the-art approaches, especially regarding prediction models on which look-ahead traffic signal control algorithms rely, so that the latest techniques are considered.

PCD Algorithm

When it comes to improving the PCD algorithm, various topics come into play, of which the computation time is undoubtedly the most important one. In reducing the computation time of the PCD algorithm, at least three conceptual solutions can be thought of:

- *Dynamic consolidation of the horizon*; by splitting up e.g. the first 20 seconds of the horizon into ten intervals of 2 seconds and splitting up the next 80 seconds according to twenty intervals of 4 seconds, a planning can be determined for a horizon of 100 seconds, while only 30 horizon intervals are considered. This greatly reduces the decision space, which in turn reduces computation time. The use of larger intervals is, however, at the expense of the accuracy of the planning. In the same context it could also be thought of to reduce the number of control options that need to be evaluated by increasing the interval size of control options (e.g. $X(s) = [t_\lambda^{\text{fg}}, t_\lambda^{\text{fg}} + 2, t_\lambda^{\text{fg}} + 4, \ldots, s]$), instead of evaluating all control options between fixed green and the end of the limited horizon (i.e. $X(s) = [t_\lambda^{\text{fg}}, \ldots, s]$);
- *Excluding stages from the optimisation process*; considering a standard four-leg intersection, one could eliminate stage $[01, 02, 03]$ from the optimisation process for smaller limited horizons if signal group 03 has no green request and no expected vehicle arrivals, since stage $[01, 02, 07, 08]$ by definition performs equally or better. For smaller limited horizons one could also exclude stages that include signal groups that have recently been green, since the PCD algorithm will likely not choose the signal groups of these stages to become green during the first part of the planning;

- *Using devices with higher computational capacity*; if, for instance, the traffic signal controller is only required to send its initial conditions and the expected vehicle arrivals to a more powerful device that is located somewhere in the cloud and that is able to run the PCD algorithm, computation time can be reduced. A reliable and fast internet connection is required for this;

Next to a reduction of the computation time, enhanced possibilities to balance between adaptability (i.e. delay reduction) and predictability could also be introduced. Currently it cannot be prevented that during recalculation of the PCD algorithm a new planning with totally different green realisations is created, which means that adaptability has a dominant position. Although there is (wide) consensus that delay reduction is the more important objective, not all road authorities may agree with this. By introducing user-adjustable penalties for time-to-green and time-to-red deviations in the PCD algorithm, the importance of deviating from an earlier planning can be taken into account, thereby increasing possibilities for user-adjustable balancing between the different objectives.

TSC Algorithm
In further perfecting the TSC algorithm, the following improvement possibilities are to be considered:

- A more appropriate strategy for picking signal groups to realise alternatively, e.g. on the basis of waiting time or the remaining time until the maximum waiting time is reached, has to potential to reduce delays. Also, user-adjustable restrictions for alternative realisations could be introduced so that road authorities have increased influence on when signal groups are allowed to realise alternatively;
- By preventing unnecessary and unwanted early ending and recurrence of green delays for vehicle can be reduced. Keeping signals green can be done by creating a more strict coupling with the planning. Here, additional attention needs to be paid to the situation of wrong vehicle arrivals, since this may have a large impact;
- Introduce the option to temporarily switch additional green time for signal groups off, so that road authorities may themselves decide on whether or not additional green time is allowed. The same effect can be achieved by introducing additional green times to be set in a variable way, so that the additional green time for signal groups can be set to zero in case specific conditions are met (e.g. priority request of conflicting signal group).

Apart from improving the performance of the algorithm, additional attention should also be paid to the algorithm's inputs. That is, TSC algorithm includes many variables (e.g. maximum waiting time, detector gap times, queue discharge rates, set of stages) and the values of these variables are currently mostly directly copied from the Golden Controller. Although this allows for fair comparison of the performance of different algorithms, the actual values that are used may not be the values that lead to an optimal performance. For instance, queue discharge rates appear to be slightly too high, which affects the accuracy of time-to-green and time-to-red predictions in a negative way.

Algorithm Evaluation
In evaluating the performance of the new traffic signal control algorithm, a wider comparison is suggested to allow for more certain performance results. This wider comparison should include different intersections and additional scenarios and should also including platooning of vehicles.

In conclusion, despite the big step that this study sets towards a new traffic signal control algorithm, there is still a long way to go before the algorithm can actually be applied in practice. Apart from all of the improvement possibilities that have been named, the significant difficulty of the computation time, which is without a doubt the biggest issue in practical application of the algorithm, remains. Given the magnitude of the computation times, it seems unavoidable to parallelise the PCD and TSC algorithms. Although at the moment it is unclear how this parallelisation should be realised, there seems to be no reason why this would not be possible from a technical point of view. Furthermore, since most Dutch traffic signal control algorithms are written in the C programming language C, the PCD and TSC algorithms will have to be converted from Python to C too.

Bibliography

L. Anderson, G. Gomes, and A.M. Bayen. Evaluation of horizontal and vertical queuing models in relation to observed trajectory data in a signalised urban traffic network. November 2014. Transportation Research Board 94th annual meeting. Washington D.C., United States of America.

R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, USA, 1957.

B. Bhowmik. Dynamic programming - it's principles, applications, strengths and limitations. *International Journal of Engineering Science and Technology*, (2(9), 2010):4822–4826, 2010.

C. Cai, C. Kwong Wong, and B.G. Heydecker. Adaptive traffic signal control using approximate dynamic programming. *Transportation Research Part C: Emerging Technologies*, (17):456–474, 2009.

K. Chandan, A.M. Seco, and A.B. Silva. Real-time traffic signal control for isolated intersection, using car-following logic under connected vehicle environment. *Transportation Research Procedia*, (25):1610–1625, 2017.

S. Chen and J. Sun. An improved adaptive signal control method for isolated signalized intersection based on dynamic programming. *IEEE Intelligent Transportation Systems Magazine*, (4), 2016.

E. Demaine. Massachusetts institute of technology open courseware. lecture 19: Dynamic programming i: Fibonacci, shortest paths. Obtained from: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/lecture-videos/lecture-19-dynamic-\programming-i-fibonacci-shortest-paths/. Accessed: 15-01-2019, 2011.

M. Dotoli, M.P. Fanti, and C. Meloni. A signal timing plan formulation for urban traffic control. *Control Engineering Practice*, (14):1297–1311, 2006.

B. Fan, D.D. Stancil, and H. Krishnan. Toward understanding characteristics of dedicated short range communications (dsrc) from a perspective of vehicular network engineers. 2010. ACM Press 329.

E. Felici. De mogelijkheden van floating car data. *Netwerk Management Magazine*, (2017-1):8–11, 2017.

Y. Feng, K.L. Head, S. Khoshmagham, and M. Zamanipour. A real-time adaptive signal control in a connected vehicle environment. *Transportation Research Part C: Emerging Technologies*, (55):460–473, 2015.

R. Florin and S. Olariu. A survey of vehicular communications for traffic signal optimization. *Vehicular Communications*, (2):70–79, 2015.

N.H. Gartner. Opac: A demand-responsive strategy for traffic signal control. *Transportation Research Record*, (906):75–81, 1983.

T. Ghose. What is the fibonacci sequence? Obtained from: https://www.livescience.com/37470-fibonacci-sequence.html. Accessed: 15-01-2019, 2018.

N.J. Goodall. *Traffic Signal Control with Connected Vehicles*. PhD thesis, University of Virginia, United States of America, May 2013.

Google. Google maps. Obtained from: https://www.google.com/maps. Accessed: 04-06-2018, 2018.

Goudappel Coffeng. C-ITS tijdvoorspellingen. Internal document, 2016a.

Goudappel Coffeng. Het gouden regelen: onze invulling van slimme verkeersregelinstallaties in 'connected mobility', 2016b.

Goudappel Coffeng. Innovatieve verkeerslichtenregelingen. Internal document, 2016c.

Goudappel Coffeng. Iver detectieconfiguratie en signaalgroepafhandeling. 2018.

Goudappel Coffeng. Analyse effecten goudenregeling n65 - helvoirt. January 2019.

V. Gradinescu, C. Gorgorin, R. Diaconescu, and V. Cristea. Adaptive traffic lights using car-to-car communication. May 2007. Vehicular Technology Conference, VTC2007-Spring. IEEE 65th.

S.I. Guler, M. Menendez, and L. Meier. Using connected vehicle technology to improve the efficiency of intersections. *Transportation Research Part C: Emerging Technologies*, (46):121–131, 2014.

F. Gündoğan. *Simplified traffic responsive signal control method for developing large cities*. PhD thesis, Graz University of Technology, Austria, September 2012.

Q. He, K.L. Head, and J. Ding. Pamscod: Platoon-based arterial multi-modal signal control with online data. *Transportation Research Part C: Emerging Technologies*, (20):164–184, 2012.

N. Helmy. An intelligent traffic flow progression model for predictive control applications. Master's thesis, Delft University of Technology, the Netherlands, December 2017.

J.J. Henry, J.L. Farges, and J. Tuffal. The prodyn real time traffic algorithm. Technical report, Département d'Etudes et de Recherches en Automatique (DERA) du Centre d'Etudes et de Recherches de Toulouse, 1983. BP 40 25, 31055, Toulouse Cedex, France.

M.R. Islam, D.S. Hurwitz, and K.L. Macuga. Improved driver responses at intersections with red signal countdown timers. *Transportation Research Part C: Emerging Technologies*, (63):207–221, 2016.

M.R. Islam, A.A. Wyman, and D.S. Hurwitz. Safer driver responses at intersections with green signal countdown timers. *Transportation Research Part F: Traffic Psychology and Behaviour*, (51):1–13, 2017.

IVER. Signaalgroepafwikkeling, detectieconfiguratie. 2002.

Y. Jiang, S. Li, and D.E. Shamo. A platoon-based traffic signal timing algorithm for major-minor intersection types. *Transportation Research Part B: Methodological*, (40):543–562, 2006.

P. Jing, H. Huang, and L. Chen. An adaptive traffic signal control in a connected vehicle environment: A systematic review. *MDPI: Information*, 2017.

D. Kari, G. Wu, and M.J. Barth. Development of an agent-based online adaptive signal control strategy using connected vehicle technology. October 2014. Proceedings of the 2014 IEEE conference on Intelligent Transportation Systems. Qingdao, China.

V.L. Knoop. Reader traffic flow theory and simulation cie4821-09. Technical report, Delft University of Technology, January 2017.

Londonist. Here's what the world's first traffic lights, in westminster, looked like. Obtained from: https://londonist.com/london/history/here-s-what-the-world-s-first-traffic-lights-in-westminster-looked-like. Accessed: 27-03-2018, 2017.

P. Mirchandani and L. Head. A real-time traffic signal control system: architecture, algorithms and analysis. *Transportation Research Part C: Emerging Technologies*, (9):415–532, 2001.

L. Misdom and G. van der Burgt. Groen op maat. *Verkeerskunde*, (06-09):39–43, 2009.

Omroep Brabant. Stoplichten n65 in vught verdwijnen: 'mooiste dag van mijn leven als wethouder'. Obtained from: https://www.omroepbrabant.nl/nieuws/2890333/Stoplichten-N65-in-Vught-verdwijnen-Mooiste-dag-van-mijn-leven-als-wethouder. Accessed: 28-11-2018, November 2018.

K. Pandit, D. Ghosal, H.M. Zhang, and C.N. Chuah. Adaptive traffic signal control with vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 2013.

Platform Beter Benutten. Talking traffic. Obtained from: https://www.beterbenutten.nl/talking-traffic. Accessed: 05-03-2018, n.d.

I. Porche and S. Lafortune. Adaptive look-ahead optimization of traffic signals. *Journal of Intelligent Transportation Systems*, (4(3)):209–254, 1997.

C. Priemer and B. Friedrich. A decentralized adaptive traffic signal control using v2i communication data. pages 1–6, October 2009. 12th International IEEE Conference on Intelligent Transportation Systems. St. Louis, MO, USA.

D.I. Robertson and R.D. Bretherton. Optimum control of an intersection for any known sequence of vehicle arrivals. September 1974. 2nd IFAC-IFIP-IFORS symposium on Traffic Control and Transportation Systems. Monte Carlo, Monaco.

L.A. Rodegerdts et al. Signalized intersections: Informational guide. Technical report, United States Department of Transportation, Federal Highway Administration, August 2004. Report No.: FHWA-HRT-04-091.

A.M. Salomons. Assignment ct4822: Simulation of an intersection. 2015.

R. Scheepjens. Algorithm design for traffic signal timings predictions of vehicle actuated controlled intersections using support vector regression. Master's thesis, Delft University of Technology, the Netherlands, May 2016.

S. Sen and L. Head. Controlled optimization of phases at an intersection. *Transportation Science*, 30(1), February 1997.

C. Stolz and B. Veroude. Nieuwe rekenmethode (voertuigafhankelijke) verkeerslichtenregelingen, November 2013. Submission for 2013 Nationaal Verkeerskundige Congres (NVC).

Trafficquest. Verkeer in nederland 2017, August 2017.

B. van der Bijl, J.Vreeswijk, J. Bie, and E. van Berkum. Car drivers' perception and acceptance of waiting time at signalized intersections. October 2011. 14th International IEEE Conference on Intelligent Transportation Systems. Washington, DC, USA.

P. van Erp, V.L. Knoop, and S. Hoogendoorn. Real-time schattingen van voertuigaccumulatie met fcd. *Netwerkmanagement Magazine*, (2):10–12, 2018.

R. van Katwijk, H. Taale, L. Prinsen, H.J. Mostert, L. Deckers, and R. Kooijman. Het nieuwe regelen: kansen en bedreigingen voor de huidige aanpak, 2016. Submission for 2016 Nationaal Verkeerskundige Congres (NVC).

R.T. van Katwijk. *Multi-Agent Look-Ahead Traffic-Adaptive Control*. PhD thesis, Delft University of Technology, the Netherlands, January 2008.

Wegenwiki. Verkeersregeling. Obtained from: `https://www.wegenwiki.nl/Verkeersregeling#Gestandaardiseerde_nummering_van_signaalgroepen`. Accessed: 29-06-2018, 2015.

A. Wilson. *Handboek Verkeerslichtenregelingen*. CROW, Ede, November 2014. ISBN 9789066286436.

R. Wu. The application of dynamic programming in production planning. May 2017. AIP Conference Proceedings 1839, 020155.

X. Zheng and W. Recker. An adaptive control algorithm for traffic-actuated signals. *Transportation Research Part C: Emerging Technologies*, (30):93–115, 2013.

A

Traffic Signal Control Basics

A.1. Signal Group

The building blocks of signalised intersections are so called *signal groups*. A signal group consists of one or more lanes that always receive the same signal indication (Gündoğan, 2012). All signal groups of an intersection are coded and in the Netherlands this coding typically takes place in clockwise order starting east. In contrast to many countries around the world, most Dutch intersections have separate facilities for less well protected road users. By spatially separating pedestrians and cyclists from motorised vehicles, traffic safety is improved. Also, by defining separate signal groups priority can be given to specific road users.

Figure A.1 presents the standard Dutch coding procedure. As can be observed, codes 01-12 are dedicated to motorised vehicles, codes 21-28 are used for cyclists, codes 31-38 are reserved for pedestrians and codes 41-52 are assigned to public transport. A wide range of additional codes is available for follow-up signal groups (e.g. 02 and 62) or when the design of the intersection is not supported by the standard coding procedure (e.g. in case of a five-leg intersection).

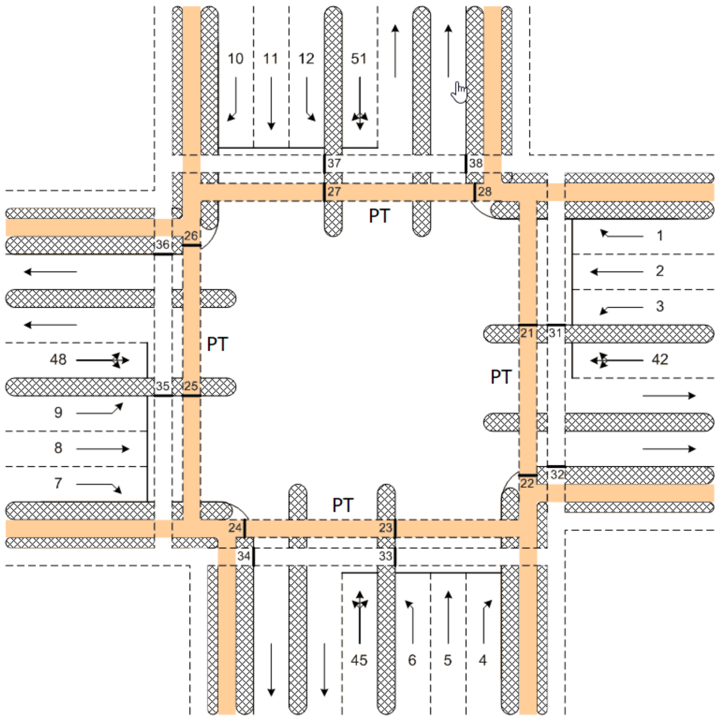


Figure A.1: Dutch coding for signal groups applied to a typical four-leg intersection. The abbreviation of PT relates to public transport (figure source: Wegenwiki, 2015).

In many cases a signal group represents a single direction, for instance the stream of traffic that passes an intersection from north to south. This is, however, no necessity. In urban areas where the room for streaming areas is often limited, a single signal group can represent up to three directions. Signal groups 42, 45, 48 and 51 in figure A.1 and signal groups 08 and 11 in figure A.2 are examples of this.

A.2. Block Procedure

Since signal groups have mutual conflicts (see Figure A.4), not all signal groups may be green at the same time. To determine when signal groups may be green, a block procedure is introduced. In a block procedure a number of *blocks* are defined and each of the blocks include several signal groups that may be green simultaneously (see Figure A.3). All signal groups of an intersection are assigned to one (or multiple) block(s). Starting with the first block, the traffic signal controller loops through the blocks to make sure all signal groups have a chance to become green at least once in a cycle. The traffic signal controller will only proceed to the next block if all signal groups in the current block have either received green and are turning amber and red, or did not receive green at all in case there was no traffic. The order in which blocks are looped through is called the *block sequence*. The time that is required for the traffic signal controller to loop through all of the blocks and return to the first block again is called the *cycle time*. Typical cycle times lay between 60 and 120 seconds, but in certain cases, for instance when public transport is prioritised, cycle times may increase greatly.

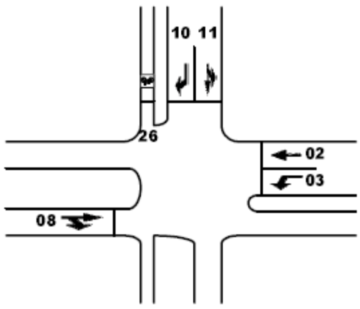


Figure A.2: Standard Dutch coding implemented on a typical intersection (figure source: van Katwijk, 2008).

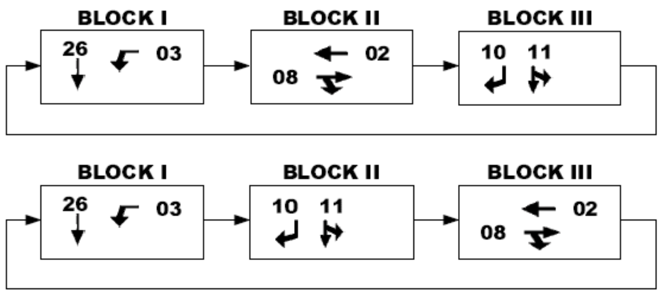


Figure A.3: Two possible block sequences for the intersection of figure A.2. As can be observed, signal groups 08 and 11 have combined directions. This is common practice for intersections with limited streaming areas (figure source: van Katwijk, 2008).

A.3. Green Realisation Types

Green of a signal group can be given on the basis of four different realisation types. To explain these types, the intersection of Figure A.2 is considered along with the block sequences of Figures A.3 and A.5.

The first realisation type, which is the standard one, is the *primary realisation*. A signal group is realised primarily if it receives green and its block is in active state. For example, if block I is currently active, then signal groups 03 and 26 may be realised primarily (see Figure A.3). In a primary realisation a signal group that is allowed to become green has the formal right on having green (Scheepjens, 2016).

The second type is the *primary ahead realisation*. A signal group is realised primary ahead when that signal group receives green while it is assigned to the next block that will become active. As an example, consider the following situation: let block I of Figure A.3 be the block that is currently active, let there be a queue of vehicles at signal group 02 and let there only be a few vehicles at signal group 08. As soon as there is no more traffic at signal group 08, green of that signal group may end and signal group 03 from block II may become green. Since signal group 02 still receives green and hence block I is still active, signal group 03 is realised primary ahead.

The third type is the alternative realisation. A signal group is realised alternatively when that signal group is neither assigned to the current active block nor to the next block that will become active, but when there is still room for its realisation (i.e. no conflicting signal group is currently green). The alternative realisation method is thereby meant to fill time gaps. Consider Figure A.5 and the following situation as an example: let block I be the block that is currently active, let there be a queue of vehicles at signal group 08 and let there be a few vehicles at signal group 02. As soon as all vehicles of signal group 02 have passed the intersection, green of that signal group may end. At that point, despite of any green requests, signal groups 03 and 26 from block II may not realise primary ahead since they have a conflict with signal group 08. However, signal group 10 from block III can be realised and subsequently that signal group may be given green alternatively if there is a green request.

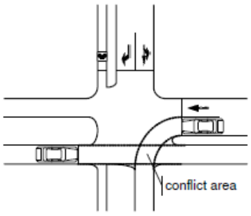


Figure A.4: A conflict area between two signal groups (figure source: van Katwijk, 2008).

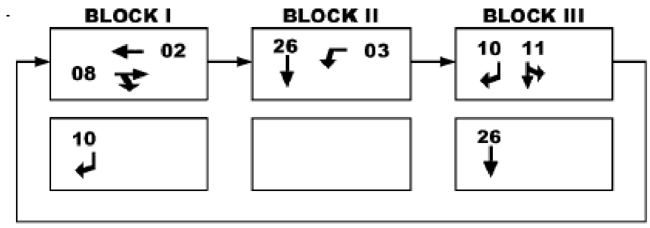


Figure A.5: Block sequence with possible alternative realisations for the intersection of figure A.2 (figure source: van Katwijk, 2008).

Additional to the above mentioned types is the priority realisation. Priority realisations are special realisations that are only available for specific road users (e.g. public transport, emergency services). These realisations have the ability to overrule any other realisation method. In case of a priority realisation, the traffic signal controller will end green of all conflicting signal groups to clear the intersection, even though these conflicting signal groups may have the formal right on green, after which the signal group that is to be realised with priority will be given green. When green of a signal group that is realised with priority ends, the signal controller will proceed with the primary realisation of the next block at the point the block sequence was interrupted.

A.4. Detection Configuration

Using infrastructure-based sensors such as cameras, radar, push buttons and inductive loop detectors, traffic signal controllers have the ability to collect real-time data of the traffic situation at the intersection. On the basis of this data control decisions can be made. The extent to which detection devices are applied greatly differs between countries and the Dutch detection configuration is considered among the most comprehensive worldwide.

The standard Dutch loop detector configuration is developed by IVER (2002) and is presented in Figure A.6. This configuration includes a stop line loop detector, an extension loop detector and one or more distant loop detectors.

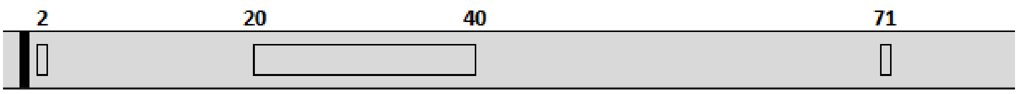


Figure A.6: Default loop detector configuration for a single lane. From left to right the stop line, extension and distant loop detectors are presented. The values represent the distance in meters between the stop line and the start/end of the loop detector. This configuration is valid for lanes with a maximum speed between 54 and 80 km/h.

Over the years various alternative configurations have been suggested. Some ten years ago Misdom and van der Burgt (2009) presented *Groen op Maat* and, more recently, Goudappel Coffeng also presented their results of a study into this subject, of which Figure A.7 includes an example (Goudappel Coffeng, 2018).

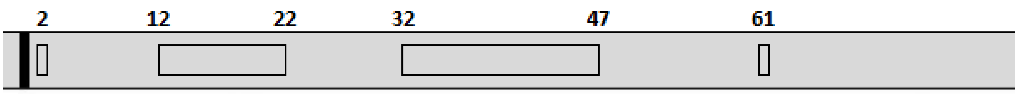


Figure A.7: An alternative loop detector configuration for a single lane. From left to right the stop line, two extension and one stop line loop detector are presented. The values represent the distance in meters between the stop line and the start/end of the loop detector. This configuration is valid for lanes with a maximum speed of 50 km/h.

A.5. Signal Completion

In the Dutch approach to traffic signal control, the green, amber and red colours that are displayed to road users via the traffic signals are actually split up into several components that are walked through sequentially (see Figure A.8). *Red before request* (RV) is the first state and applies to the time period wherein a signal group is not allowed to become green. In this state it does not matter whether or not green is requested. The state switches to *red after request* (RA) when a green request has been set and when the traffic signal controller has decided that a signal group may become green. A signal group in the RA state will receive green as soon as the clearance time of all conflicting signal groups have elapsed and in case there is no conflicting signal group that is to be realised with priority. As soon as the signal group's signal changes to green the *fixed green time* (FG) starts. Fixed green times are typically set to 5 seconds. When the fixed green time has ended, a signal group may come into *waiting green* (WG) if there are no green requests for conflicting signal groups. In any other case, green progresses into *extension green* (VG). In the extension green state the signal group receives green until some criteria are no longer met. These criteria typically include a combination of detector occupancy, gap times and a predefined maximum green time. After extension green the state progresses into *prolonging green* (MG). In this state green is realised, independent of green requests, as long as it does not prevent other signal groups that do have green requests from realising. In the final state green turns *amber* (GL) for a fixed amount of time and the process repeats again by starting at the RV state.

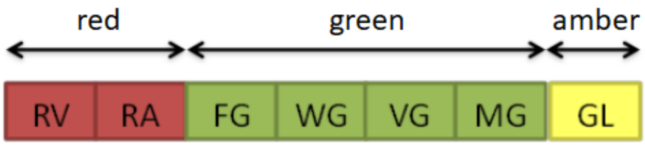


Figure A.8: Signal completion as a function of all its different components (figure source: van Katwijk, 2008).

A.6. Control Strategies

On the basis of the block procedure and the signal completion a number of different traffic signal control strategies can be defined. Despite the great variety of names that is used in literature (see Appendix C), three types of traffic signal control strategies are generally distinguished. The next paragraphs describe the characteristics of these strategies and mention some of their benefits and limitations.

A.6.1. Fixed-time Traffic Signal Control

A fixed-time traffic signal control strategy is characterised by fixed green times, a fixed cycle duration and a static block sequence. This means that signal groups receive green in a predefined and definite order and green times do not depend on the number of vehicles present at the intersection. Green is also realised in case there are no vehicles at all. Fixed-time traffic signal controllers have a simplified signal completion as only fixed green (FG), amber (GL) and red before request (RV) exists. Furthermore, signal groups are only realised primarily. The timing plans of fixed-time traffic signal controllers are created on the basis of historical data and often there exist a number of timing plans for different times of the day to address demand fluctuation.

Although this type of traffic signal controller has certain advantages over other types such as a simple design, a high predictability (i.e. every cycle is exactly the same and is repeated continuously) and lows costs (there is no need for detection elements), fixed-time traffic signal control is not commonly applied in the Netherlands. The main reason for this is that this type of strategy is unable to handle varying amounts of traffic well. Consequently, fixed-time traffic signal control often leads to a lot of unnecessary waiting time. Furthermore, credibility is well-known problem for this type of traffic signal controller since road users typically do not understand why they have to wait for a red signal when there is no vehicle activity at the intersection. Fixed-time signal control is, however, commonly applied in other parts of the world, such as in the United States of America (Florin and Olariu, 2015).

A.6.2. Actuated Traffic Signal Control

Actuated traffic signal controllers collect real-time traffic data via infrastructure-based sensors and apply simple logic to make control decisions. In actuated traffic signal control the block sequence is fixed, but green times vary depending on the presence of traffic (Stolz and Veroude, 2013). That is, green of a signal group may terminate prematurely in case there are no more vehicles, or green can be extended up to a predefined maximum when there is a large queue. As a result, the cycle time of the traffic signal controller is variable. Also, in actuated traffic signal control green is only realised when there are vehicles that profit from this, hence signal groups that have no green request are skipped. Actuated traffic signal control includes primary, primary ahead and alternative green realisations and all of the states of signal completion. Actuated traffic signal control is widely spread throughout the Netherlands and they make up about 85% of all traffic signal controllers (van Katwijk, 2008).

Thanks to their adaptability to real-time traffic conditions, actuated traffic signal control is especially useful in off-peak periods with low traffic volumes and in situations where traffic flows vary a lot. Compared to fixed-time traffic signal control, actuated traffic signal controllers manage to greatly decrease delays as a result of this flexibility. The flexibility that actuated traffic signal control offers, however, also leads to low predictability of control decisions, hence presenting reliable time-to-green and time-to-red predictions to road users is difficult.

A.6.3. Adaptive Traffic Signal Control

Similar to actuated traffic signal control, adaptive traffic signal controllers also rely on measurements of current traffic conditions. This type of traffic signal controller, however, offers more room for seeking optimal control decisions as they are not based on data from individual signal groups only, but rather on traffic data that has been obtained from all signal groups of the intersection. For instance, an actuated traffic signal controller may decide to terminate or extend green of a particular signal group on the basis of that signal group's loop detectors, while an adaptive traffic signal controller may decide to change the order in which signal groups receive green to generate a better fit with respect to overall delay minimisation. Furthermore, adaptive traffic signal controllers may utilise real-time traffic data to predict near future traffic conditions and seek optimal control decisions accordingly. Look-ahead traffic signal control as proposed in this thesis falls within this category.

Thanks to their increased flexibility and their decision-making on the basis of greater amounts of traffic data, adaptive traffic signal control strategies often allow for an even greater reduction of delays when compared to actuated traffic signal control. On the downside, predictability of the signal controller's cycle is typically even more difficult. Also, many adaptive traffic signal control strategies represent a high level of complexity and engineers typically require months to understand them (Goodall, 2013), especially in case of optimisation of a network of intersections. Furthermore, most adaptive traffic signal control algorithms require excessive maintenance. As a result, adaptive traffic signal control is not commonly applied throughout the world.

B

Traffic Signal Control Terminology

This appendix includes a list definitions and English-Dutch translations of traffic signal control terminology that is used in this thesis.

English	*Dutch*	*Definition / Explanation*
Alternative realisation	Alternatieve realisatie	Realisation method; a signal group is realised alternatively when it receives green while it is neither assigned to the current active block nor to the next block that will become active
Block	Blok	A set of signal groups that may receive green simultaneously
Block sequence	Blokkenvolgorde	The order in which blocks are looped through in the traffic signal control algorithm
Clearance time	Ontruimingstijd	The time that is required to clear the conflict area of two conflicting signal groups after green is ended for the first signal group and green starts for the second signal group
Cycle time	Cyclustijd	The time that the traffic signal controller requires to loop through all of the blocks in the traffic signal control algorithm
Distant loop detector	Verweglus	Part of the standard Dutch detection configuration; an inductive loop detector of approximately 1 meter length located far away (typically >50 meters) from the stop line that is used to request and extend green
Extension green (VG)	Verlenggroen	Part of the signal completion; in this state green is extended until some criteria are met. These criteria are typically a combination of the predefined maximum green time and gap time
Extension loop detector	Lange lus	Part of the standard Dutch detection configuration; an inductive loop detector of approximately 20 meters length located 20-40 meters before the stop line that is used to request and extend green
Fixed green	Vastgroen	Part of the signal completion; in this state green is given for a fixed amount of time, typically 4-5 seconds
Gap time	Hiaattijd	The gap time of a loop detector denotes the elapsed time since that loop detector was last occupied
Maximum green time	Maximum groentijd	A predefined value that dictates the maximum amount of time a signal group may receive green in a single realisation. A maximum green time is defined to prevent infinite extension of green in case of for instance a traffic jam
Minimum green time	Minimum groentijd	A predefined value that dictates the minimum amount of time a signal group is to receive green in a single realisation. A minimum green time is defined to prevent flickering of signals

Passenger car equivalent (pce)	Personenauto-equivalent	The passenger car equivalent value of a mode of transport denotes the impact that that mode has on traffic variables such as headway, velocity and density compared to a single car. The pce value for a lorry is typically considered 3-4, while the pce value for a motorcycle is only 0.5
Primary ahead realisation	Versnelde primaire realisatie	Realisation method; a signal group is realised primary ahead if that signal group receives green while it is assigned to the next block that will become active
Primary realisation	Primaire realisatie	Realisation method; a signal group is realised primarily if it receives green and its block is in active state
Priority realisation	-	Realisation method; a signal group is realised with priority when it receives green while overruling the green realisation of other, conflicting signal groups. Priority realisations are typically only reserved for public transport and emergency services
Prolonging green (MG)	Meeverlenggroen	Part of the signal completion; in this state green is extended for as long as no con that are not in prolonging green
Queue discharge rate	Afrijcapaciteit	Indicates the amount of traffic that is able to pass through a signal if it turns green for one hour. The queue discharge rate is a function of many different variables such as the lane width, lane sloop and weather conditions and typical values lay in the range of 1700-2100 pae/hour (Gündoğan, 2012)
Red after request (RA)	Rood na aanvraag	Part of the signal completion; in this state a green request is set and the signal group will become green as soon as the clearance times of all conflicting signal groups have elapsed and when there is no priority request of a conflicting signal group
Red before request (RV)	Rood voor aanvraag	Part of the signal completion; in this state a signal group is not allowed to become green, independent of any green request
Signal completion	Signaalgroepafwikkeling	The collection of all internal green, red and amber states of a traffic signal controller
Signal group	Signaalgroep	A set of one or more lanes that always receive the same signal indication. In practice the term 'movement' (NL: richting) is also used for this
Signal group-based approach	-	In a signal group-based approach signal groups in blocks are individually considered for their possibility to become green
Stage	-	A set of signal groups that may receive green simultaneously
Stage-based approach	-	In a stage-based approach stages with signal groups are considered as a single unit, rather than considering individual signal groups
Stop line loop detector	Korte lus	Part of the standard Dutch detection configuration; an inductive loop detector with a length of 1 meter, typically located one meter before the stop line, that is used to request and extend green
Time-to-green	Tijd tot groen	The time-to-green time for a signal group denotes the expected time until the signals of that signal group change to green
Time-to-red	Tijd tot rood	The time-to-red time for a signal group denotes the expected time until the signals of that signal group change to red
Waiting green (WG)	Wachtgroen	Part of the signal completion; in this optional state green is given as long as there are no green requests for conflicting signal groups

C

Traffic Signal Control Strategies

A look into literature shows that there are a number of different ways to categorise signal controllers. In the next paragraphs the main types of signal controllers are set forth and their principles and limitations are mentioned.

Wilson (2014), a well-known Dutch publication, categorises fixed-time signal controllers, vehicle-actuated signal controllers and semi-actuated signal controllers. Signal controllers that fall within the first category have timing plans that are set to static schedules regardless of traffic demand. Signal controllers of the second category base their timing plans on measurements of the actual traffic situation, given a set of predefined signal control parameters (e.g. green extension up to a certain maximum green time). Semi-actuated signal control is a combination of the two where certain signal control parameters are fixed (e.g. cycle time, phase sequence), while others are variable (e.g. green time). Jiang et al. (2006) name a similar categorisation.

Zheng and Recker (2013) also distinguish three types of signal controllers, namely pre-timed, traffic-actuated and traffic-responsive/adaptive. The signal controllers of the first category are similar to the aforementioned fixed signal controller. The second category includes semi-actuated and fully-actuated and both of the types that are mentioned by Wilson (2014) can be assigned to this category. Signal controllers of the third category are so called third generation urban traffic control systems and, just like the second category, they rely on measurements of the current state of traffic. Although both actuated and adaptive/responsive signal control use real-time traffic data, they are different in two ways. First, the degree of freedom in terms of choosing optimal values for signal control parameters is greater for the third category. That is, within traffic-actuated signal control among other things cycle length and green times may differ, but changes herein are subject to a set of predefined and fixed signal control parameters (e.g. maximum green time, fixed phase sequence). By calculating the optimal value for signal control parameters, adaptive/response signal controllers have more freedom. Second, in adaptive/responsive signal control the traffic state at the entire intersection is taken into account, instead of measurements from single movements only in actuated signal control. Similar types of signal controllers are named by van Katwijk (2008), Goodall (2013), Guler et al. (2014), Feng et al. (2015), Chandan et al. (2017) and Jing et al. (2017).

Different from the aforementioned categorisations, Cai et al. (2009) distinguish offline and online signal control methods. Offline methods use historical traffic data as input for the calculation of signal plans, and can therefore be compared to fixed-time control strategies. Online methods include signal controllers that use traffic information that is collected real- time via, for instance, loop detectors to develop responsive signal control. The latter category includes responsive control (use real-time data to calculate up-to-date signal settings) and plan selection (select a preset signal plan that fits best). A somewhat similar categorisation is mentioned by Dotoli et al. (2006) too.

Despite all the different terms that are used in literature, there seems to be a clear distinction between signal controllers that do and do not use real-time traffic data to base signal timings on; online versus offline or fixed-time methods. Furthermore, within online signal control one could distinguish adaptive and actuated signal controllers, whereas the difference between the two relates to the degree of freedom in terms of choosing/ calculating optimal signal control parameters and the scale in which measurements of traffic are taken into account. Consequently, three types of signal control are distinguished in this thesis, namely fixed-time, actuated and adaptive.

D
Algorithm Variables

Variable	*Domain*	*Unit*	*Description*
$A_{\lambda,k}$	$\mathbb{N}_0$	veh	The number of vehicles that are expected to arrive at signal group λ at time k
$A_{\lambda,k}^{\text{composition}}$	-	-	The composition of the vehicle that are expected to arrive at signal group λ at time interval k. Each of the entries in this vector represents the type of vehicle. Example: $[Car, Car, Bus, Car]$. By definition the number of elements in this vector equals $A_{\lambda,k}$
$A_{\lambda,i}^{\text{fraction}}$	-	-	The fraction of the vehicles that are expected to arrive at signal group λ at time interval k. Each of the entries in this vector represents the fraction of a vehicle that is still present. By definition each entry in this vector equals 1 and the number of elements equals $A_{\lambda,k}$. Example: $[1,1,1,1]$
$A_{\lambda,i}^{\text{weight}}$	-	-	The weight factor of the vehicles that are expected to arrive at signal group λ at time interval k. Each of the entries in this vector represents the weight of a vehicle. Example: $[1,1,3,1]$. By definition the number of elements in this vector equals $A_{\lambda,k}$
$\alpha_{\lambda,k}$	-	-	This boolean denotes whether or not signal group λ has a green request at time k
$\beta_{\lambda,k}$	-	-	This boolean denotes whether or not signal group λ is blocked from becoming green at time k
$c_{\lambda}^{\text{initial}}$	-	-	The queue composition at the start of recalculation of the PCD algorithm. Each of the entries in this vector represents the type of vehicle. Example: $[Car, Car, Bus, Car]$
$c(s,x,j,\lambda,k)$	-	-	The queue composition of signal group λ in iteration j at time interval k given a limited horizon s and a control decision x. Each of the entries in this vector represents the type of vehicle. Example: $[Car, Car, Bus, Car]$
$C_{s,j,\lambda}^{*}$	-	-	The queue composition of signal group λ that corresponds to the best control decision for limited horizon s in iteration j. Each of the entries in this vector represents the type of vehicle. Example: $[Car, Car, Bus, Car]$
CP^{duration}	-	-	A vector containing the optimal duration of stages. Example: $[8,13,5]$
CP_{λ}^{ge}	-	-	A vector containing the green end times for signal group λ according to the optimal planning of control decisions. Example: $[9,35]$
CP_{λ}^{gs}	-	-	A vector containing the green start times for signal group λ according to the optimal planning of control decisions. Example: $[4,28]$
CP^{stages}	-	-	A vector containing the optimal sequence of stages. Example: $[B,A,C]$
$\gamma_{\lambda,k,d}$	-	-	This boolean denotes whether or not detector d of signal group λ is occupied at time k
$d(s,x,j,\lambda,k)$	$\mathbb{N}_0$	s	The delay of signal group λ in iteration j for time interval k given a limited horizon s and a control decision x
$D_{s,j,\lambda}^{*}$	$\mathbb{N}_0$	s	The delay for signal group λ that corresponds to the best control decision for limited horizon s in iteration j
FC	-	-	A user-defined ordered set of signal groups

FC^{rog}	-	-	A set of signal groups $\lambda \in FC$ that are included in the planning of control decisions. The set is ordered according to the green start time of signal groups
$ge(s,x,j,\lambda)$	$\mathbb{Z}$	s	The green end time of signal group λ in iteration j given a limited horizon s and control decision x
$ge_{\lambda}^{\text{initial}}$	$\mathbb{Z}$	s	The green end time of signal group λ at the start of recalculation of the PCD algorithm
$GE^{*}_{s,j,\lambda}$	$\mathbb{Z}$	s	The green end time of signal group λ that corresponds to the best control decision for limited horizon s in iteration j
$gs(s,x,j,\lambda)$	$\mathbb{Z}$	s	The green start time of signal group λ in iteration j given a limited horizon s and control decision x
$GS^{*}_{s,j,\lambda}$	$\mathbb{Z}$	s	The green start time of signal group λ that corresponds to the best control decision for limited horizon s in iteration j
h	$\mathbb{N}_1$	s	Full planning horizon length
h^{bt}	$\mathbb{N}_0$	s	Backtrack horizon. This value indicates how many time intervals from h remain to be included into the optimal control policy
j	$\mathbb{N}_1$	-	Iteration counter of the PCD algorithm
j^{*}	$\mathbb{N}_1$	-	The final iteration j that was performed in the PCD algorithm
j^{min}	$\mathbb{N}_1$	-	The absolute minimum number of iterations that the PCD algorithm has to perform to find the best control policy
k	$\mathbb{N}_1$	s	Time index
λ	$\mathbb{N}_1$	-	Signal group index
μ_{λ}	$\mathbb{N}_1$	veh/h	The queue discharge rate for signal group λ
N^{stages}	$\mathbb{N}_1$	-	The number of stages in the ordered set *SG*. This variable hence denotes the cardinality of *SG*
φ	$\mathbb{N}_1$	-	Stage index
$q_{\lambda}^{\text{initial}}$	-	-	The queue fraction of signal group λ at the start of recalculation of the PCD algorithm. Each of the entries in this vector represents the fraction of a vehicle that is still present in the queue. Example: $[0.4, 1, 1, 1]$
$q(s,x,j,\lambda,k)$	-	-	The queue fraction of signal group λ in iteration j at time interval k given a limited horizon s and a control decision x. Each of the entries in this vector represents the fraction of a vehicle that is still present in the queue. Example: $[0.4, 1, 1, 1]$
$Q^{*}_{s,j,\lambda}$	-	-	The queue fraction of signal group λ that corresponds to the best control decision for limited horizon s in iteration j. Each of the entries in this vector represents the fraction of a vehicle that is still present in the queue. Example: $[0.4, 1, 1, 1]$
s	$\mathbb{N}_1$	s	Limited horizon
$s_{\lambda,k}^{\text{colour}}$	-	-	The colour that the signals of signal group λ display to road users at time k. This colour can either be *red*, *amber* or *green*
$s_{\lambda,k}^{\text{status}}$	-	-	The internal status of signal group λ at time k. This status can either be *RV*, *RA*, *FG*, *VG*, *MG* or *GL*
SG_{φ}	-	-	A user-defined ordered set of stages. Stage φ denotes a selection of signal groups λ of set *FC*
t	$\mathbb{R}$	s	Real time
$t^{\text{cl}}_{\lambda,\lambda^{*}}$	$\mathbb{Z}$	s	The clearance time between signal groups λ and λ^{*}. A clearance time of -99 denotes that there is no conflict between signal groups λ and λ^{*}
t_{λ}^{fg}	$\mathbb{N}_1$	s	The fixed green time of signal group λ. The fixed green time represents the minimum green time
t_{λ}^{ga}	$\mathbb{N}_0$	s	The time that signal group λ may be given green additional to the green time that is incorporated for signal group λ in the planning of control decisions
$t^{\text{gap}}_{\lambda,k,d}$	$\mathbb{R}$	s	The gap time for detector d of signal group λ at time k. The gap time represents the elapsed time since the detector was last occupied
$t^{\text{gap}^{\text{set}}}_{\lambda,k,d}$	$\mathbb{R}$	s	The set gap time for detector d of signal group λ at time k. The set gap time represents the time that a detector may be unoccupied before any action such as ending green may take place
$t^{\text{gd}}_{\lambda,k}$	$\mathbb{R}$	s	The elapsed time at time k since the signal(s) of signal group λ turned green. This value is zero if the signal(s) of signal group λ are currently not green
t_{λ}^{ge}	$\mathbb{N}_0$	s	The latest moment at which green of signal group λ must end
t_{λ}^{gs}	$\mathbb{N}_0$	s	The earliest moment at which green of signal group λ can start

$t^{\text{gp}}_{\lambda,k}$	$\mathbb{R}$	s	The remaining green time for signal group λ at time k according to the planning of control decisions
$t^{\text{rd}}_{\lambda,k}$	$\mathbb{R}$	s	The elapsed time at time k since the signal(s) of signal group λ turned red. This value is zero if the signal(s) of signal group λ are currently not red
$t^{\text{recalculate}}_{k}$	$\mathbb{R}$	s	The remaining time at time step k before a new planning of control decisions is determined
t^{rg}_{λ}	$\mathbb{N}_1$	s	The guaranteed red time of signal group λ
t^{vg1}_{λ}	$\mathbb{R}$	s	The maximum extension green time of signal group λ on the basis of the stop line, extension and distant loop detectors
t^{vg2}_{λ}	$\mathbb{R}$	s	The maximum extension green time of signal group λ on the basis of the extension and distant loop detectors
$t^{\text{wa}}_{\lambda,k}$	$\mathbb{R}$	s	The waiting time of signal group λ at time k, hence the elapsed time since the signal(s) of signal group λ turned red while there has been a green request. This value is zero if the signal(s) of signal group λ are currently not red
t^{wm}_{λ}	$\mathbb{N}_1$	s	The maximum allowed waiting time of signal group λ
t^{y}_{λ}	$\mathbb{N}_1$	s	The set amber time of signal group λ
$t^{\text{yd}}_{\lambda,k}$	$\mathbb{R}$	s	The elapsed time at time k since the signal(s) of signal group λ turned amber. This value is zero if the signal(s) of signal group λ are currently not amber
T	$\mathbb{R}$	s	Model time step size
$TTG^{\text{certainty}}_{\lambda,k}$	-	-	The time-to-green certainty indication for signal group λ at time k. This certainty indication equals either '=' or '<>', which denotes a certain or uncertain time-to-green prediction respectively and which depends on whether or not signal group λ is included in the planning of control decisions
$TTG^{\text{penalty}}_{\lambda,k}$	$\mathbb{N}_0$	s	The summed change of the time-to-green prediction for signal group λ at time k. A change results from a new planning being determined in which signal group λ is planned to become green at a different moment in time
$TTG^{\text{time}}_{\lambda,k}$	$\mathbb{N}_0$	s	The time-to-green prediction for signal group λ at time k
$TTR^{\text{certainty}}_{\lambda,k}$	-	-	The time-to-red certainty indication for signal group λ at time k. This certainty indication equals either '=' or '<>', which denotes a certain or uncertain time-to-green prediction respectively and which depends on whether or not signal group λ is included in the planning of control decisions
$TTR^{\text{penalty}}_{\lambda,k}$	$\mathbb{N}_0$	s	The summed change of the time-to-red prediction for signal group λ at time k. A change results from a new planning being determined in which signal group λ is planned to become red at a different moment in time
$TTR^{\text{time}}_{\lambda,k}$	$\mathbb{N}_0$	s	The time-to-red prediction for signal group λ at time k
v	-	-	Detector index
$V^{\text{type}}_{\lambda,k}$	-	-	The type of vehicle that occupies the distant loop detector of signal group λ at time k
$V^{\text{weight}}_{\lambda,k}$	$\mathbb{R}$	-	The weight of the vehicle that occupies the distant loop detector of signal group λ at time k
$w^{\text{initial}}_{\lambda}$	-	-	The queue weight of signal group λ at the start of recalculation of the PCD algorithm. Each of the entries in this vector represents the weight of a vehicle. Example: $[1,1,3,1]$
$w(s,x,j,\lambda,k)$	-	-	The queue weight of signal group λ in iteration j at time interval k given a limited horizon s and a control decision x. Each of the entries in this vector represents the weight of a vehicle. Example: $[1,1,3,1]$
$W^{*}_{s,j,\lambda}$	-	-	The queue weight of signal group λ that corresponds to the best control decision for limited horizon s in iteration j. Each of the entries in this vector represents the weight of a vehicle. Example: $[1,1,3,1]$
wf_{λ}	$\mathbb{R}$	-	The weight factor for signal group λ
x	$\mathbb{N}_0$	s	Control option; x denotes the number of time intervals that are allocated to a stage
$X(s)$	-	-	An ordered set of feasible control options x given a limited horizon s
$X^{*}_{s,j}$	$\mathbb{N}_0$	s	The number of time intervals that are allocated to a stage in the best control policy for limited horizon s in iteration j

z	$\mathbb{N}_0$	s	The number of time intervals that is left to allocate to the stage or combination of stages that form the best control policy for the corresponding limited horizon in the previous iteration $j-1$
$\omega_\lambda^{\text{distance}}$	$\mathbb{N}_1$	meter	The distance between the distant loop detector and the stop line of signal group λ
$\omega_\lambda^{\text{velocity}}$	$\mathbb{N}_1$	km/h	The average velocity with which vehicles at signal group λ typically travel between the distant loop detector and the stop line

E
TTG and TTR Accuracy Distributions

The figures in this appendix present distributions of time-to-green and time-to-red accuracy. A negative TTG accuracy denotes that green started later than planned. A positive TTG accuracy denotes that that green started earlier than planned. A negative TTR accuracy denotes that red started later than planned and a positive TTR accuracy denotes that red started earlier than planned.

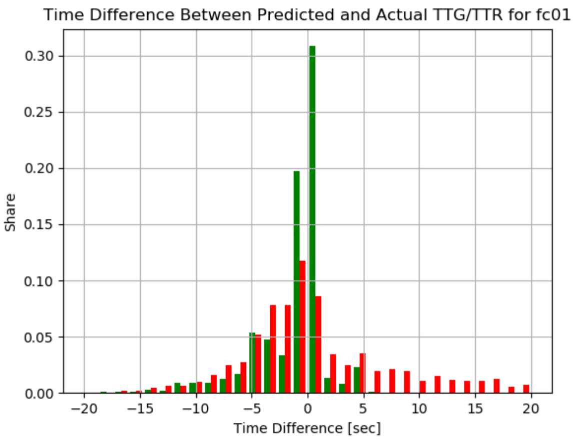


Figure E.1: The TTG and TTR accuracy distribution for signal group 01 in scenario 5 (equal flows, no vehicle arrivals).

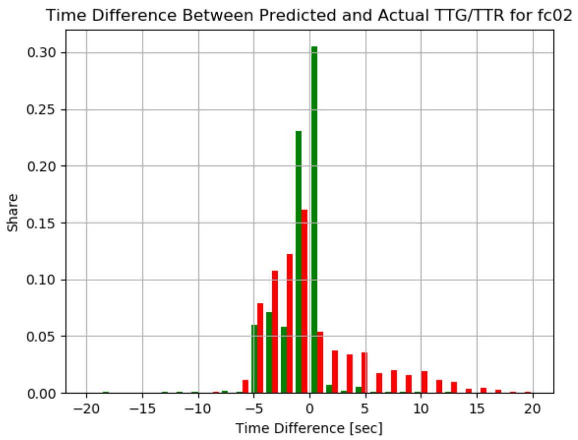


Figure E.2: The TTG and TTR accuracy distribution for signal group 02 in scenario 5 (equal flows, no vehicle arrivals).

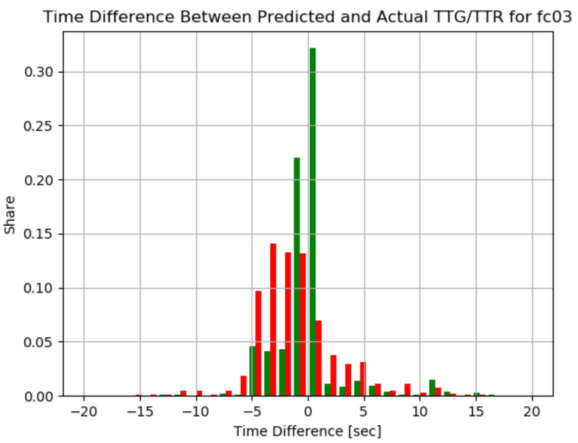


Figure E.3: The TTG and TTR accuracy distribution for signal group 03 in scenario 5 (equal flows, no vehicle arrivals).

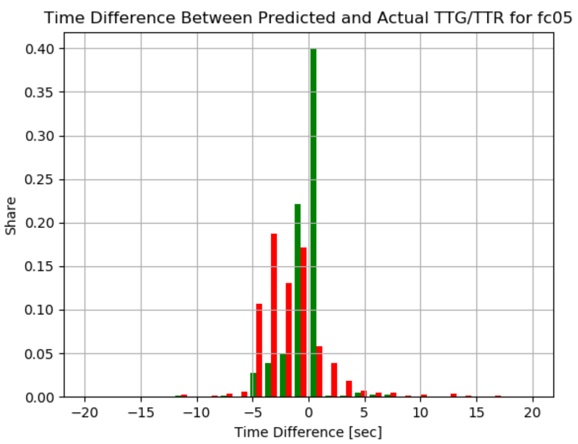


Figure E.4: The TTG and TTR accuracy distribution for signal group 05 in scenario 5 (equal flows, no vehicle arrivals).

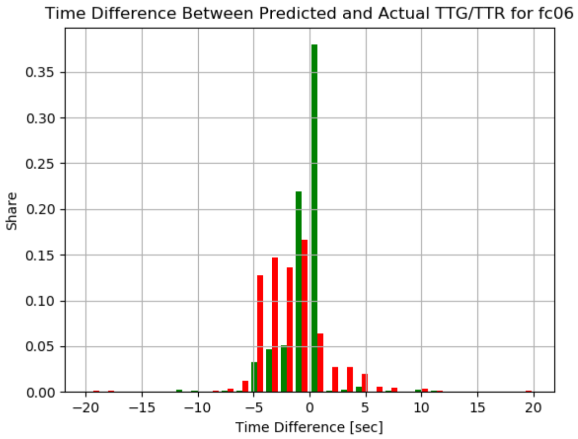


Figure E.5: The TTG and TTR accuracy distribution for signal group 06 in scenario 5 (equal flows, no vehicle arrivals).

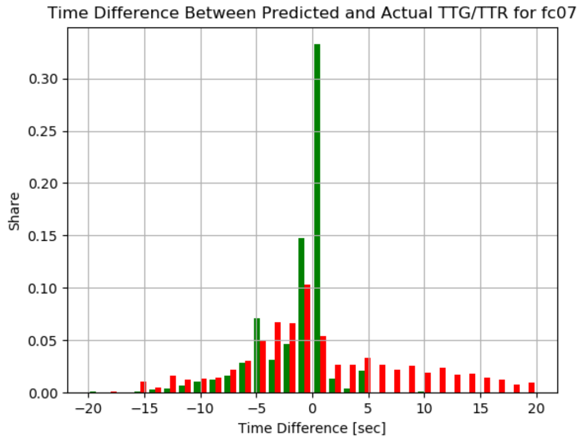


Figure E.6: The TTG and TTR accuracy distribution for signal group 07 in scenario 5 (equal flows, no vehicle arrivals).

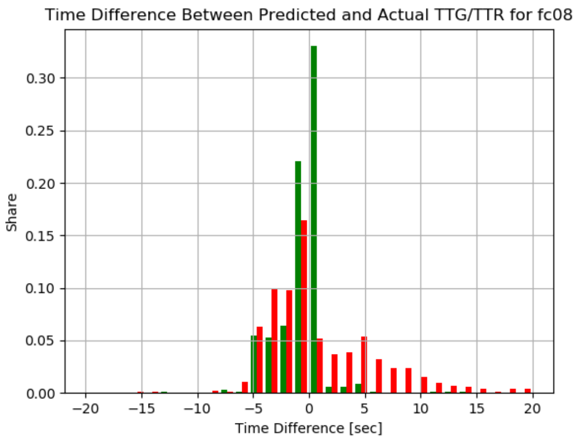


Figure E.7: The TTG and TTR accuracy distribution for signal group 08 in scenario 5 (equal flows, no vehicle arrivals).

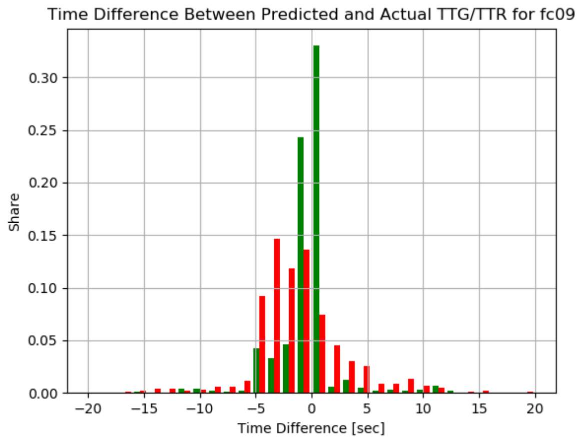


Figure E.8: The TTG and TTR accuracy distribution for signal group 09 in scenario 5 (equal flows, no vehicle arrivals).

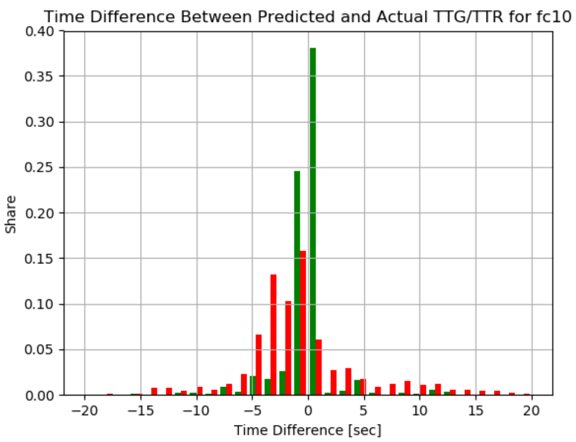


Figure E.9: The TTG and TTR accuracy distribution for signal group 10 in scenario 5 (equal flows, no vehicle arrivals).

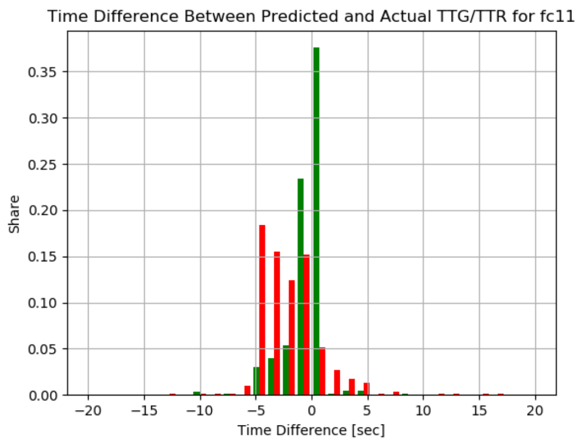


Figure E.10: The TTG and TTR accuracy distribution for signal group 11 in scenario 5 (equal flows, no vehicle arrivals).

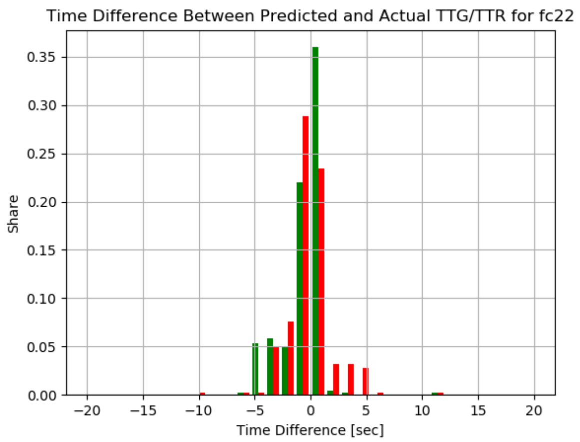


Figure E.11: The TTG and TTR accuracy distribution for signal group 22 in scenario 5 (equal flows, no vehicle arrivals).

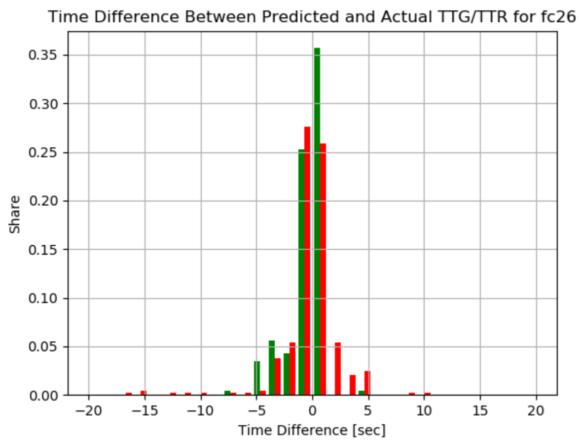


Figure E.12: The TTG and TTR accuracy distribution for signal group 26 in scenario 5 (equal flows, no vehicle arrivals).

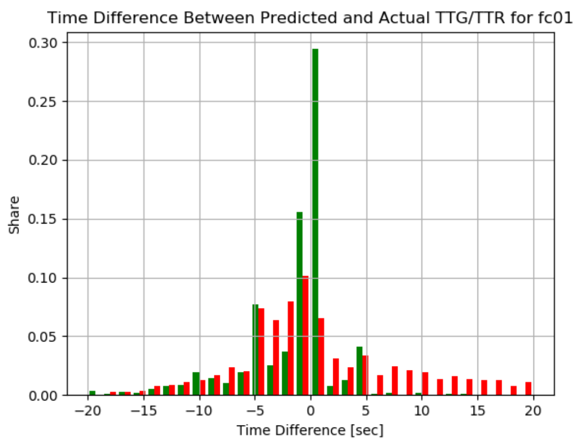


Figure E.13: The TTG and TTR accuracy distribution for signal group 01 in scenario 6 (major-minor, no vehicle arrivals).

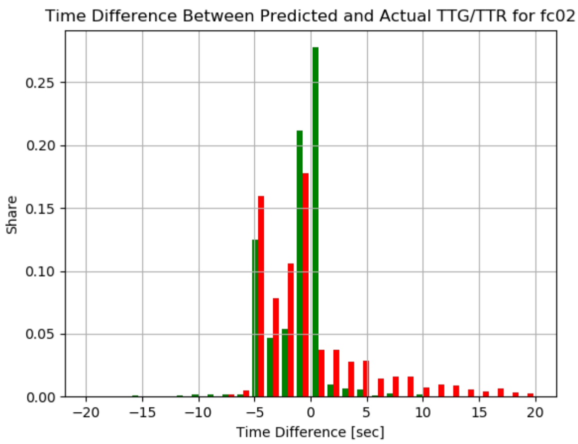


Figure E.14: The TTG and TTR accuracy distribution for signal group 02 in scenario 6 (major-minor, no vehicle arrivals).

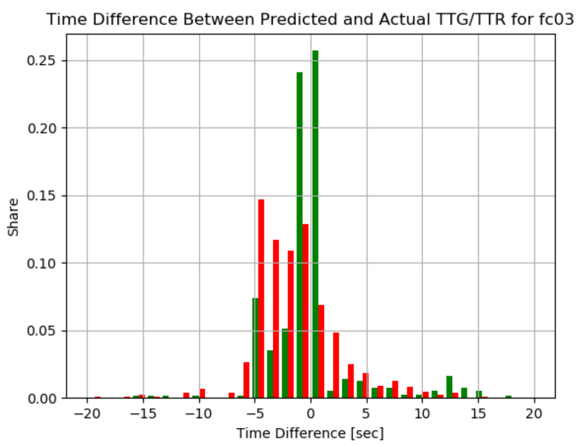


Figure E.15: The TTG and TTR accuracy distribution for signal group 03 in scenario 6 (major-minor, no vehicle arrivals).

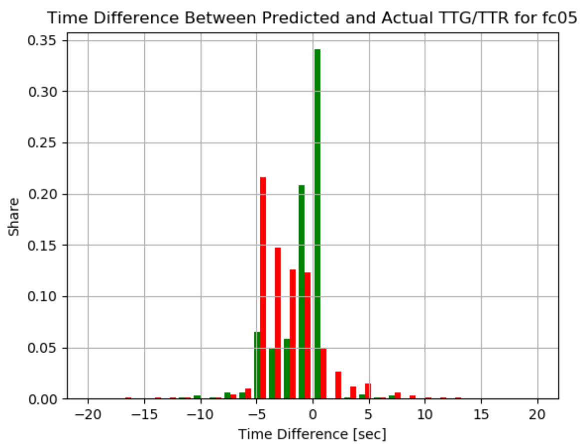


Figure E.16: The TTG and TTR accuracy distribution for signal group 05 in scenario 6 (major-minor, no vehicle arrivals).

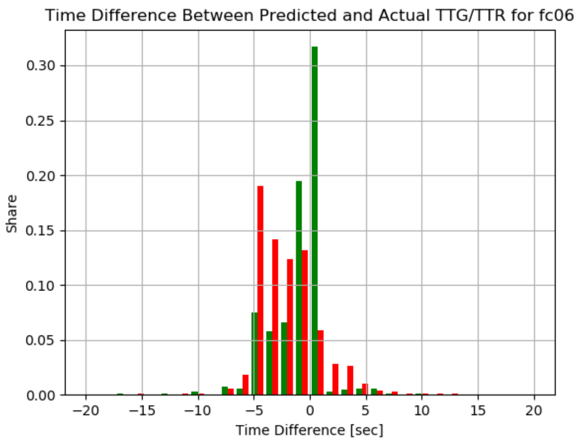


Figure E.17: The TTG and TTR accuracy distribution for signal group 06 in scenario 6 (major-minor, no vehicle arrivals).

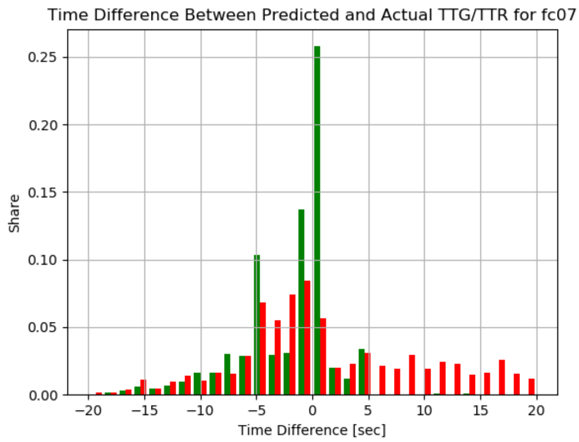


Figure E.18: The TTG and TTR accuracy distribution for signal group 07 in scenario 6 (major-minor, no vehicle arrivals).

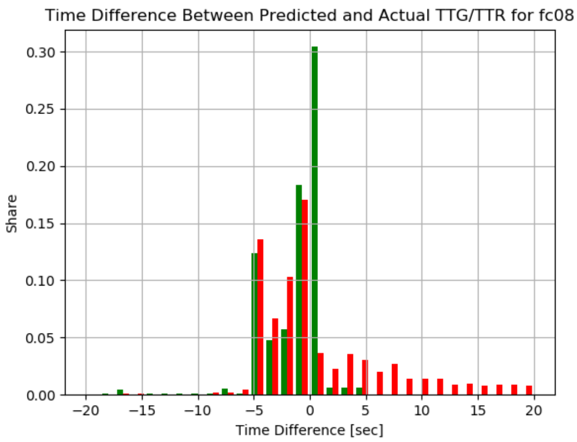


Figure E.19: The TTG and TTR accuracy distribution for signal group 08 in scenario 6 (major-minor, no vehicle arrivals).

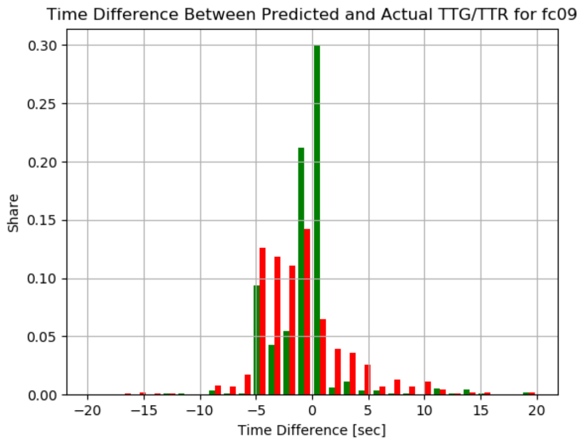


Figure E.20: The TTG and TTR accuracy distribution for signal group 09 in scenario 6 (major-minor, no vehicle arrivals).

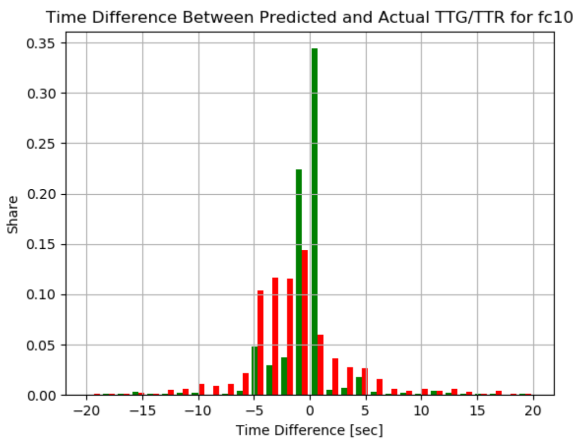


Figure E.21: The TTG and TTR accuracy distribution for signal group 10 in scenario 6 (major-minor, no vehicle arrivals).

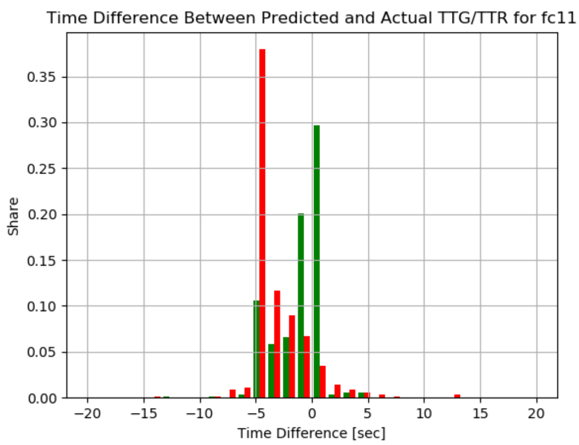


Figure E.22: The TTG and TTR accuracy distribution for signal group 11 in scenario 6 (major-minor, no vehicle arrivals).

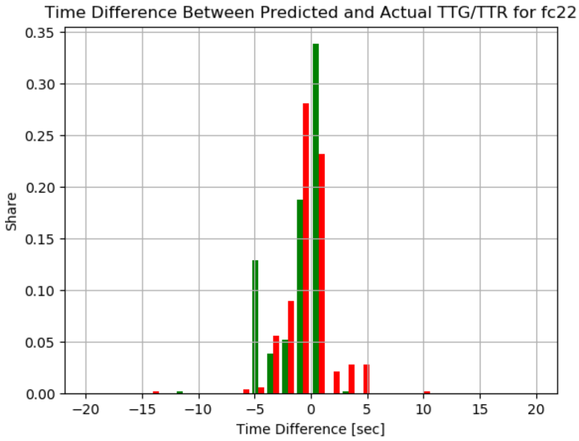


Figure E.23: The TTG and TTR accuracy distribution for signal group 22 in scenario 6 (major-minor, no vehicle arrivals).

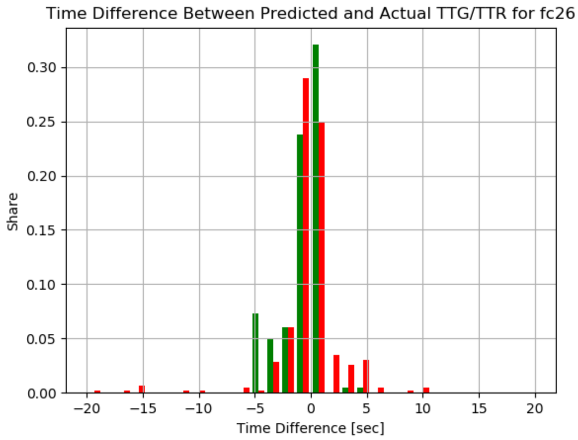


Figure E.24: The TTG and TTR accuracy distribution for signal group 26 in scenario 6 (major-minor, no vehicle arrivals).

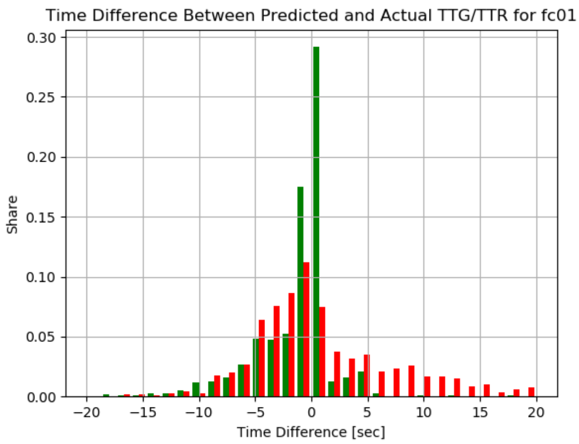


Figure E.25: The TTG and TTR accuracy distribution for signal group 01 in scenario 7 (equal flows, vehicle arrivals).

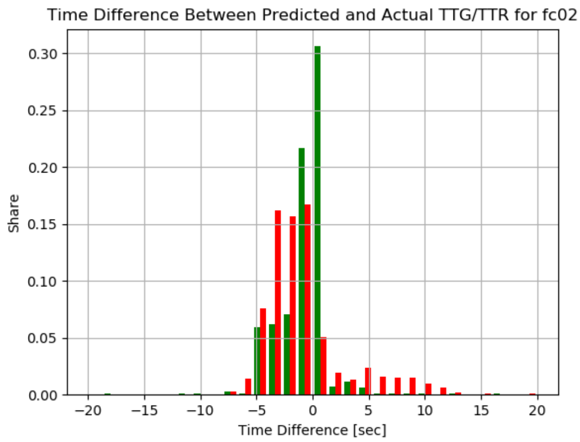


Figure E.26: The TTG and TTR accuracy distribution for signal group 02 in scenario 7 (equal flows, vehicle arrivals).

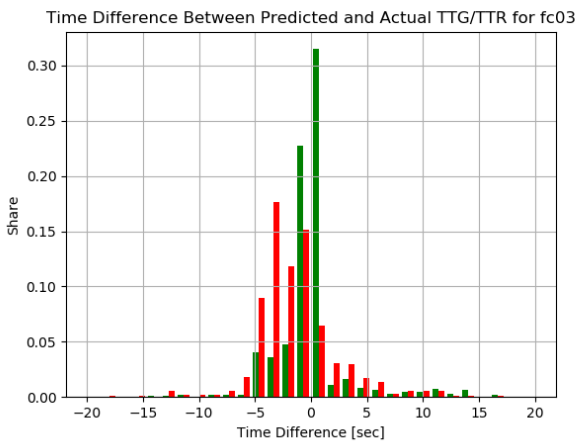


Figure E.27: The TTG and TTR accuracy distribution for signal group 03 in scenario 7 (equal flows, vehicle arrivals).

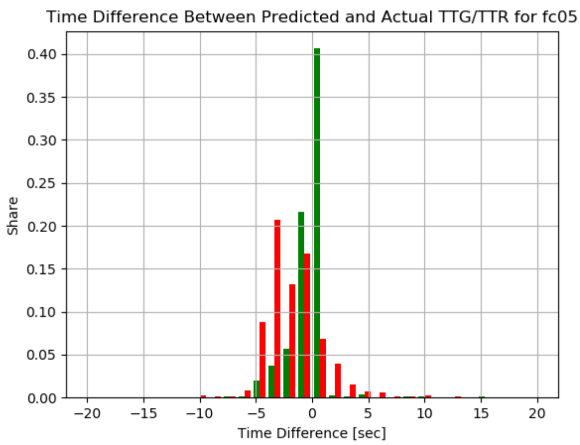


Figure E.28: The TTG and TTR accuracy distribution for signal group 05 in scenario 7 (equal flows, vehicle arrivals).

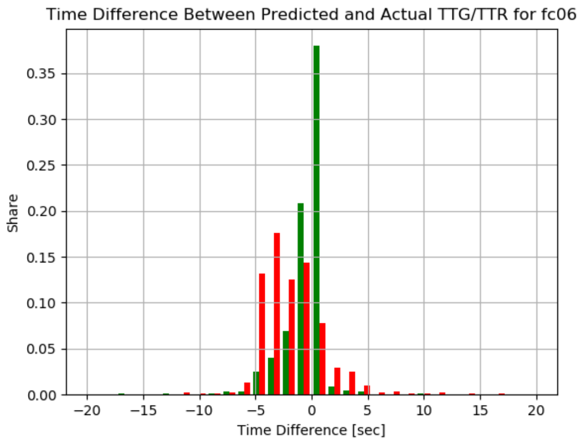


Figure E.29: The TTG and TTR accuracy distribution for signal group 06 in scenario 7 (equal flows, vehicle arrivals).

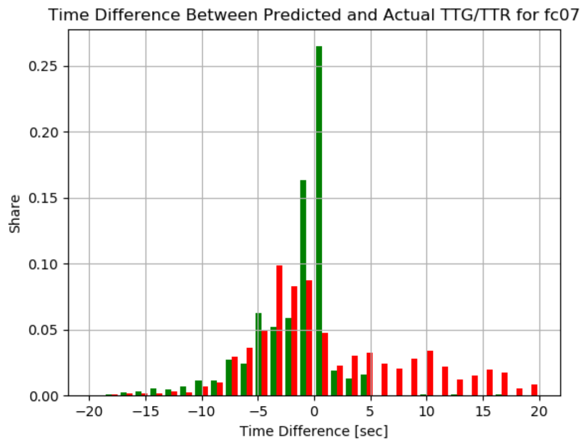


Figure E.30: The TTG and TTR accuracy distribution for signal group 07 in scenario 7 (equal flows, vehicle arrivals).

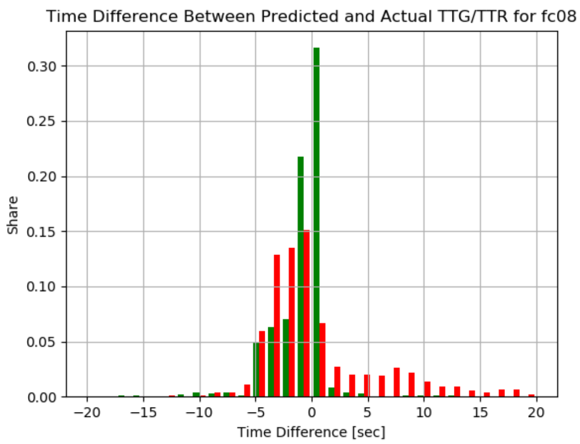


Figure E.31: The TTG and TTR accuracy distribution for signal group 08 in scenario 7 (equal flows, vehicle arrivals).

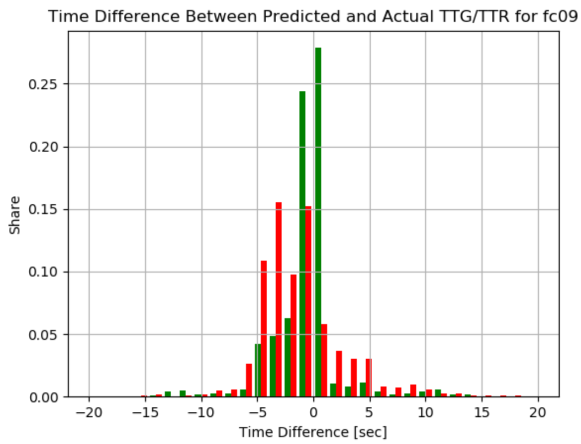


Figure E.32: The TTG and TTR accuracy distribution for signal group 09 in scenario 7 (equal flows, vehicle arrivals).

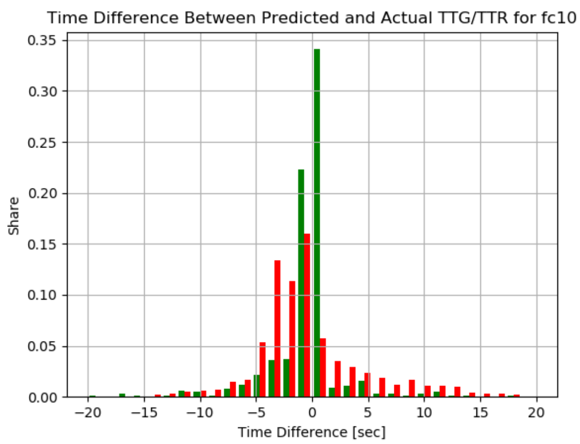


Figure E.33: The TTG and TTR accuracy distribution for signal group 10 in scenario 7 (equal flows, vehicle arrivals).

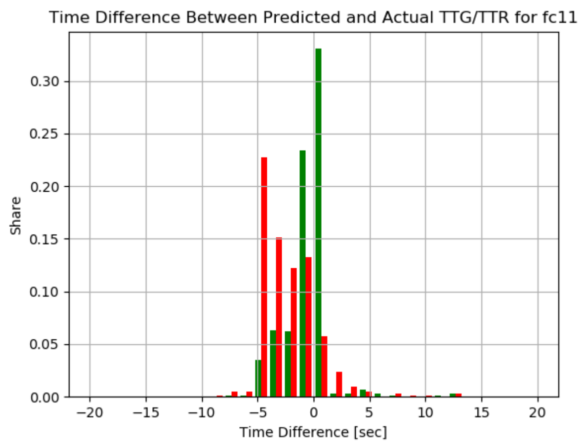


Figure E.34: The TTG and TTR accuracy distribution for signal group 11 in scenario 7 (equal flows, vehicle arrivals).

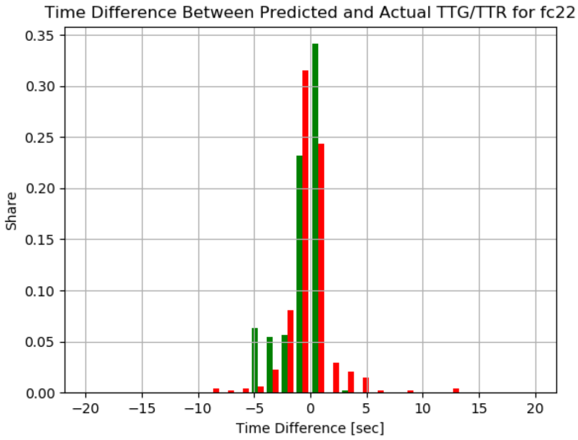


Figure E.35: The TTG and TTR accuracy distribution for signal group 22 in scenario 7 (equal flows, vehicle arrivals).

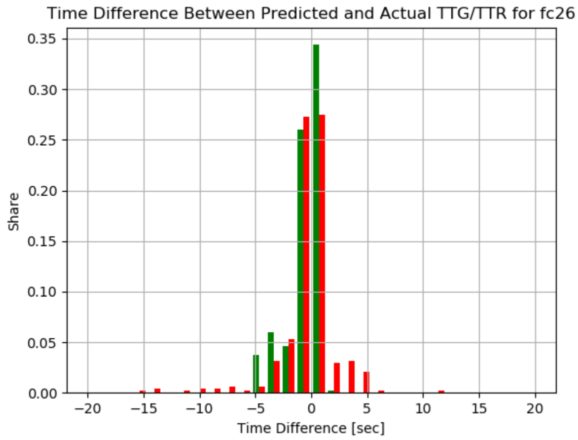


Figure E.36: The TTG and TTR accuracy distribution for signal group 26 in scenario 7 (equal flows, vehicle arrivals).

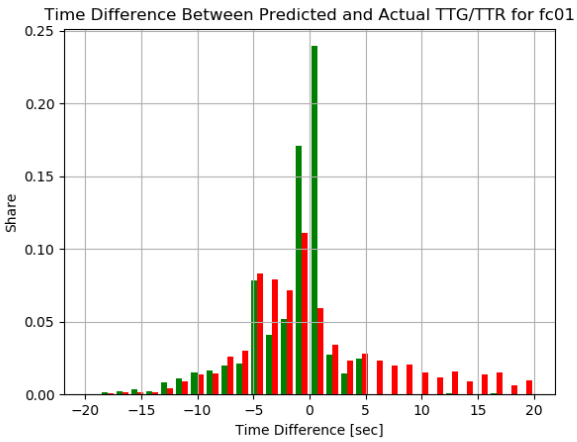


Figure E.37: The TTG and TTR accuracy distribution for signal group 01 in scenario 8 (major-minor, vehicle arrivals).

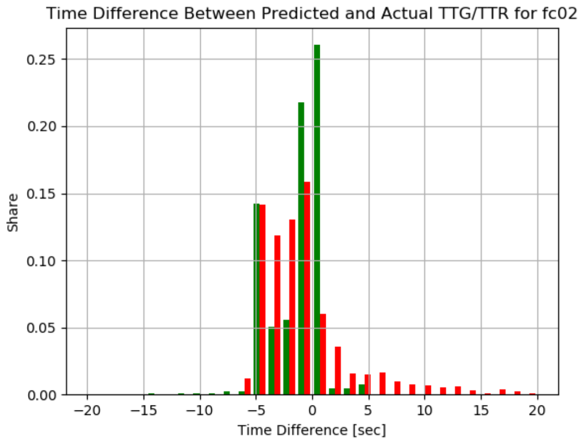


Figure E.38: The TTG and TTR accuracy distribution for signal group 02 in scenario 8 (major-minor, vehicle arrivals).

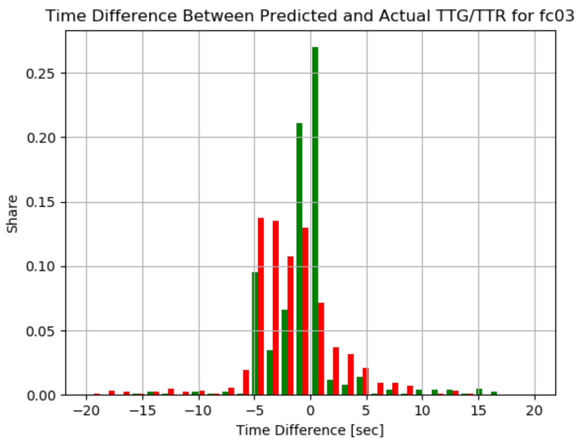


Figure E.39: The TTG and TTR accuracy distribution for signal group 03 in scenario 8 (major-minor, vehicle arrivals).

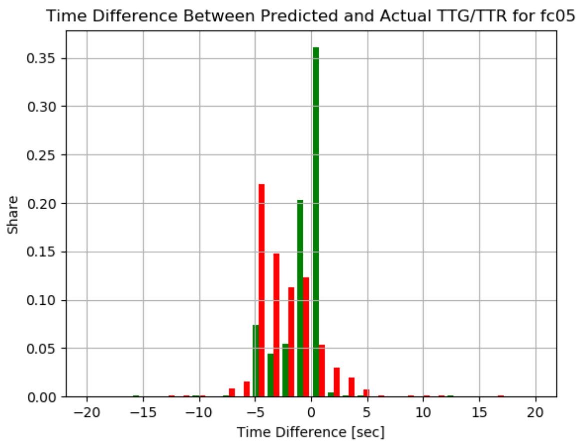


Figure E.40: The TTG and TTR accuracy distribution for signal group 05 in scenario 8 (major-minor, vehicle arrivals).

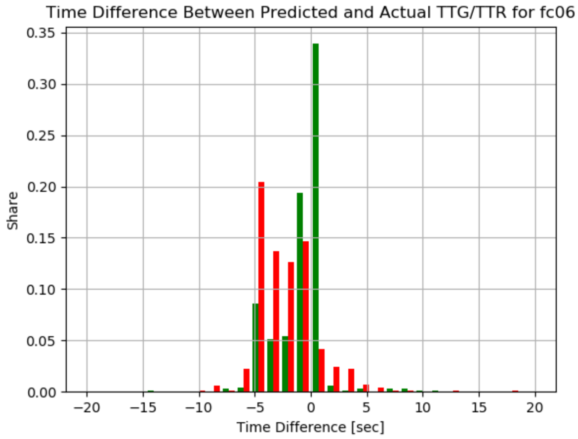


Figure E.41: The TTG and TTR accuracy distribution for signal group 06 in scenario 8 (major-minor, vehicle arrivals).

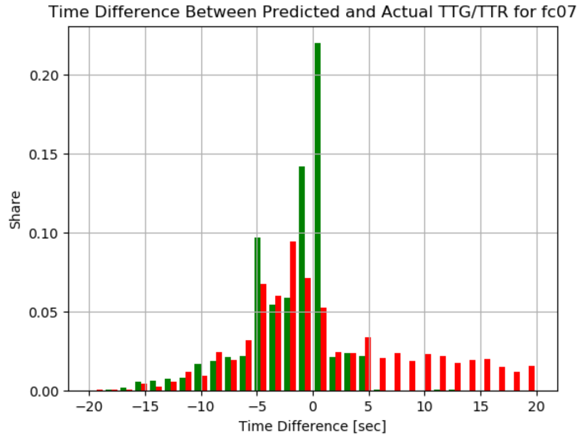


Figure E.42: The TTG and TTR accuracy distribution for signal group 07 in scenario 8 (major-minor, vehicle arrivals).

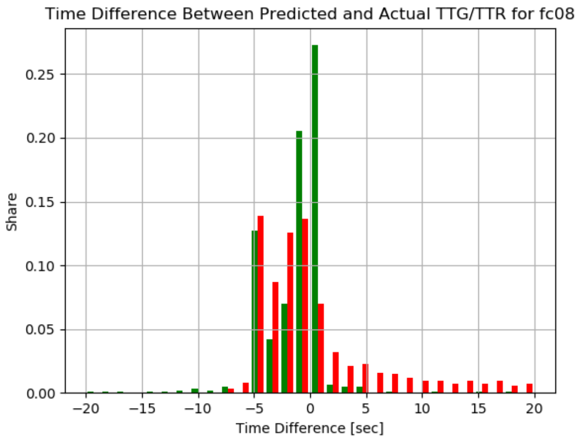


Figure E.43: The TTG and TTR accuracy distribution for signal group 08 in scenario 8 (major-minor, vehicle arrivals).

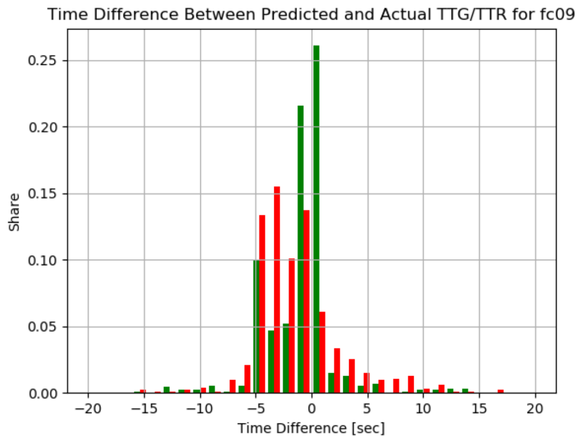


Figure E.44: The TTG and TTR accuracy distribution for signal group 09 in scenario 8 (major-minor, vehicle arrivals).

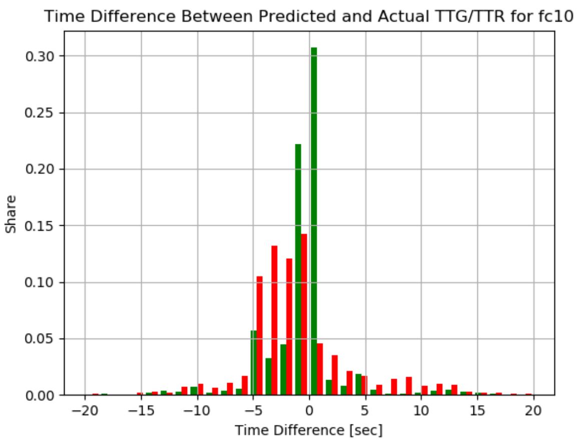


Figure E.45: The TTG and TTR accuracy distribution for signal group 10 in scenario 8 (major-minor, vehicle arrivals).

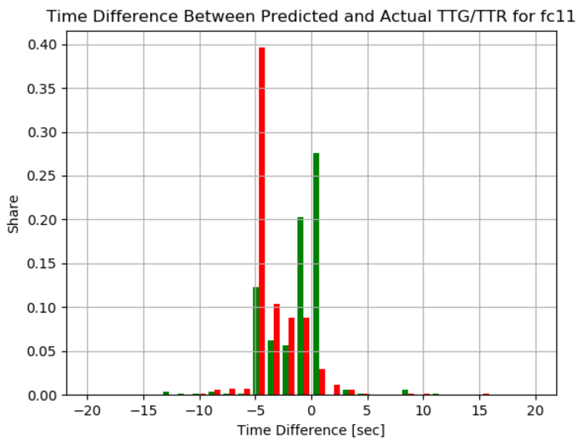


Figure E.46: The TTG and TTR accuracy distribution for signal group 11 in scenario 8 (major-minor, vehicle arrivals).

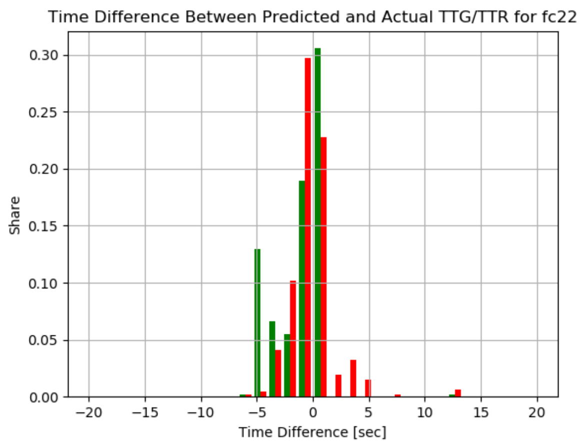


Figure E.47: The TTG and TTR accuracy distribution for signal group 22 in scenario 8 (major-minor, vehicle arrivals).

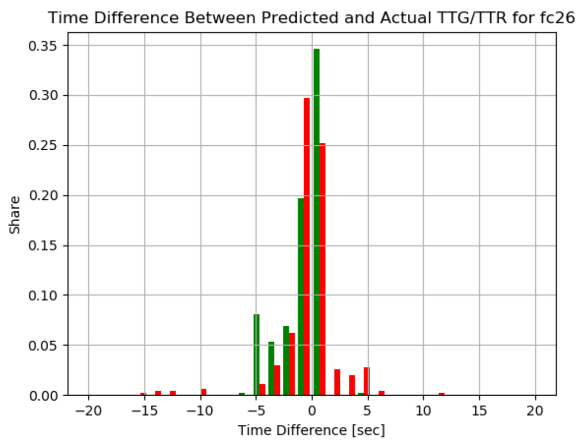


Figure E.48: The TTG and TTR accuracy distribution for signal group 26 in scenario 8 (major-minor, vehicle arrivals).